FINITE DIFFERENCE SOLUTIONS FOR PLATE BUCKLING PROBLEMS

MAXIMO S. DUMLAO, JR.

519

FINITE DIFFERENCE SOLUTIONS FOR PLATE BUCKLING PROBLEMS

by

Maximo S. Dumlao, Jr.
Lieutenant, Philippine Navy
B. S., Philippine Military Academy, 1959

Submitted in partial fulfillment
for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

UNITED STATES NAVAL POSTGRADUATE SCHOOL
May 1966

NPS Archive
1966
Dunlap, M.

ABSTRACT

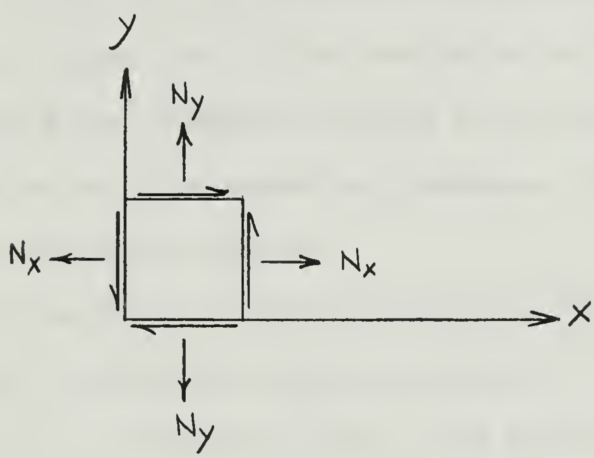
A digital computer program was developed to find the buckling coefficient for rectangular plates with all edges simply supported or with all edges clamped. A finite difference technique is used to solve the partial differential equation for the deflection of a plate classically treated as having only a small deflection compared to its thickness. The program was prepared to take for an input the stresses at nodes formed by grids dividing the plate into rectangles. The stresses and deflections at each node are used in writing difference equations. An extrapolation formula is featured in the program which allows a close approximation to the buckling coefficient without necessitating the use of a large number of grid nodes. The program was written in FORTRAN 60 but must be used as a FORTRAN 63 program to take advantage of some of its inherent flexibilities. Information is provided in the output of the program which aids in evaluating the reliability of a solution.

TABLE OF CONTENTS

LIST OF SYMBOLS

A, B, C	coefficient matrices
$D = Eh^3/12(1 - \nu^2)$	flexural rigidity of the plate
E	modulus of elasticity
F_x, F_y, F_{xy}	specified functions of x and y for expressing N_x, N_y, N_{xy} in terms of $\overline{N}$
H_x	mesh length
H_y	mesh width
K	computed buckling coefficient
$K_1 = K_{corrected}$	extrapolated buckling coefficient
K_2, K_3, K_4, K_5, K_6	constants used in extrapolation formula
$\overline{N}$	load parameter
N_x, N_y, N_{xy}	forces per unit width in middle plane of plate

Sign convention



a	plate length
b	plate width
h	plate thickness
w	plate deflection or column vector of plate deflections
x,y	rectangular coordinates

ϵ	estimated error
λ	an eigenvalue
ν	Poisson's ratio

1. Introduction.

The determination of the initial buckling load of homogeneous plates has been a subject of interest for many years. A very good compilation of past approaches may be found in the work by Gerard and Becker [8]. The methods used were varied and, in general, they may be classified into two categories: (1) variational methods (commonly known as energy methods when applied to mechanics), the best known of which are those attributed to Ritz and B. G. Galerkin and (2) numerical methods in which finite difference approximations to the partial differential equations of the deflection of the plate at a sufficient number of points on the plate result in a set of algebraic equations. The methods under the first category result in approximating an infinite set of infinite series equations by a finite set of equations which must be solved simultaneously. The second method involves the coefficient matrix of a set of algebraic equations whose eigenvalue is found or its determinant evaluated for a given value of an appropriate parameter. The parameter is varied until the determinant vanishes.

The first method results in a solution limited to the problem for which it was developed. Further, it does not promise to be simple for cases which may be more on the practical side. The second method was shown to require a considerable amount of labor and time in evaluating the determinant of a large matrix, even for a simple case.[1] Computerization suggests itself as a remedy to such limitations encountered in both

[1]A conclusion by J. Yardley in his thesis "Applications of Finite Difference Equations to Buckling Problems of Rectangular Plates", Washington University, (1948).

methods.

With this in view, a digital computer program was developed to find the buckling coefficient for rectangular plates. The program was written for plates with all edges simply supported or all edges clamped. In writing the program the governing partial differential equation for the deflection of a rectangular plate based on the classical treatment of a thin plate having only a small deflection was used. Hence, all problems that are solved using the program will also be based on the same treatment.

There are two steps that must be accomplished to get the initial buckling load. First, the governing partial differential equation is replaced by a set of linear algebraic equations by approximating it with difference quotients term by term. Second, the highest eigenvalue of the coefficient matrix of the set of equations must be found. The buckling coefficient is inversely proportional to the eigenvalue of the matrix. This will be shown in the following section.

2. Mathematical Basis.

The governing partial differential equation of the deflection of a plate in equilibrium in the absence of body forces and which is under the action of forces in its middle plane is[2]

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{1}{D}\left(N_x \frac{\partial^2 w}{\partial x^2} + N_y \frac{\partial^2 w}{\partial y^2} + 2N_{xy}\frac{\partial^2 w}{\partial x \partial y}\right) \quad (1)$$

[2]Timoshenko, S. P., and J. M. Gere, Theory of Elastic Stability (New York, Toronto, London: McGraw-Hill Book Company, 1961), p. 348.

where

w = lateral deflection of the plate

x,y = cartesian coordinates. These are conveniently taken parallel to the sides of the plate.

D = flexural rigidity of the plate

N_x = the normal force per unit width in the x-direction

N_y = the normal force per unit width in the y-direction

N_{xy} = the shearing force per unit width in the x and y directions

The forces N_x, N_y, N_{xy} may be expressed in terms of a common parameter $\overline{N}$. Thus we put

$$N_x = \overline{N}\, F_x(x,y)$$

$$N_y = \overline{N}\, F_y(x,y)$$

$$N_{xy} = \overline{N}\, F_{xy}(x,y)$$

where F_x, F_y, and F_{xy} are specified functions of x and y. Assuming that such a relationship exists, the problem of finding the buckling coefficient becomes a problem of finding the smallest value of $\overline{N}$ that would cause the plate to start to buckle, or the critical value of $\overline{N}$. To do this using finite difference technique, Eqn. 1 must be replaced first with a set of linear algebraic equations.

The plate is divided into integral parts in the x and y directions so that rectangular meshes of uniform size are formed. This manner of dividing the plate is advantageous because it is not limited by the aspect ratio of the plate and thus makes programming easier. At each corner, or "node", of the meshes a finite difference equation is written in approximation to Eqn. 1. The set of algebraic equations formed may be written in matrix form as

$$A^2 w = kBw \qquad (2)$$

where

A^2 = square coefficient matrix for the left side of Eqn. 1.

B = square coefficient matrix for the right side of Eqn. 1.

w = column vector of the plate lateral deflection at nodes.

k = function of the parameter $\overline{N}$.

Eqn. 2 may be rewritten as

$$(C - \frac{1}{k} I)w = 0 \tag{3}$$

where

$C = A^{-2}B$

I = the identity matrix of the same order as C.

Except for the trivial case when w is zero, Eqn. 3 is true only when

$$\frac{1}{K} = \lambda$$

where λ is an eigenvalue of the matrix C. Since we are seeking the smallest value of k, the highest eigenvalue λ must be found. The detailed mathematical formulation for the matrices is discussed in Appendix I.

It was realized that theoretically the outlined procedure of finding the buckling coefficient will give a value close to the correct buckling coefficient only when the number of points used is large. There are a number of objections to satisfying such a requirement. Dealing with a large number of points will require a large amount of computer storage since for a given square matrix a storage equal to the square of its order is required. The second objection to the use of a large number of points in the plate is that too much computer time will be involved. The last objection is that there is possibility of round-off which would introduce an error unless computation using double precision arithmetic

is resorted to.

To eliminate the necessity of using a large number of points on the plate an extrapolation formula is used. In doing this one effectively looks for an estimate of the error inherent in the finite difference approximations of the governing partial differential equation when using a finite number of points in the plate. In other words the inherent error is the result of replacing an infinitesimal quantity with a finite one. For this purpose the error ϵ was assumed to take the form

$$\epsilon = K_2 H_x^2 + K_3 H_x^4 + K_4 H_y^2 + K_5 H_y^4 + K_6 H_x^2 H_y^2$$

Extrapolation was achieved by solving simultaneously for K_1 six equations of the form

$$K = K_1 + K_2 H_x^2 + K_3 H_x^4 + K_4 H_y^2 + K_5 H_y^4 + K_6 H_x^2 H_y^2$$

where

K = the computed buckling coefficient for a given choice of mesh

K_1 = the extrapolated value of the buckling coefficient

K_2, K_3, K_4, K_5, K_6 = constants

H_x = the length of the mesh used

H_y = the width of the mesh

A more detailed discussion of the estimate of the error and extrapolation for the buckling coefficient is given in Appendix I.

3. Cases Considered.

Several problems in plate buckling with known solutions were solved using the program in order to compare the results that were obtained with it. The discussion below will indicate the extent of agreement with

previous solutions. All cases will be discussed with reference to tables which are the final output of the program. Tables I-VII may be found at the end of this section on pages 21-27. The information provided by the tables is

Aspect Ratio = length/width

K_1 = extrapolated value of the buckling coefficient.

K_2, K_3, K_4, K_5, K_6 = constants calculated for correcting K.

K = the computed value of the buckling coefficient

$K_{corrected}$ = corrected value of K.

K_1 and $K_{corrected}$ should agree very closely if round-off errors are not excessive since they are both based on the same equations. All quantities are dimensionless.

The analytical solutions to the problems that were considered were cast in the form

$$N_{cr} = K' \pi^2 D/b^2$$

where

b = width of the plate

K' = a dimensionless constant

In the program K was computed when the above formula is written in the form

$$N_{cr} = KD/b^2$$

Comparison of solutions will be made between K_1 and $K'\pi^2$.

CASE I. <u>Simply Supported Plate Under Uniform Compression in the X-Direction</u>.

For this case the common parameter is $\bar{N} = N_x$ so that

$$F_x = 1$$

$$F_y = F_{xy} = 0$$

The formula for the critical load of rectangular plates with aspect ratio = a/b where a is the length of the plate and b is its width is[3]

$$N_{cr} = \frac{\pi^2 D}{b^2}\left(\frac{a}{b} + \frac{b}{a}\right)^2$$

For a/b = 3/4 we have

$$N_{cr} = 42.836826\ D/b^2$$

Entering Table I we get

$$K_1 = 42.836819$$

Comparing this with the coefficient of the theoretical critical load we get a

Difference = .000016%

This small difference substantiates the form of the extrapolation formula assumed.

That the finite difference solution is accurate in this case may be shown also by considering the eigenvectors using the set of points when there are 10 divisions in the x-direction and 12 divisions in the y-direction. It is known that the deflection surface of the buckled plate may be represented by the equation[4]

$$w = a_{11} \sin(\pi x/a) \sin(\pi y/b) \qquad (4)$$

where

$$a_{11} = \text{constant}$$

Referring to the figure in Table I and setting the deflection $w_1 = 1$ at point #1, a_{11} can be determined. We get

$$a_{11} = 12.503206$$

[3]Ibid., p.353.

[4]Ibid., p. 327.

With a_{11} known we get for point #46

$$w_{46} = 12.503206$$

The computer solution for the eigenvectors gives for point #46

$$w_{46}\text{, computer} = 12.502909$$

Comparing the values obtained we have

$$\text{Difference} = .0023\%$$

CASE II. Simply Supported Plate Subjected to Pure Bending.

The figure in Table II shows the plate under pure bending load. The parameter chosen is the maximum stress designated as $\bar{N}$ in the figure so that

$$F_x = 1 - 2y/b$$

$$F_y = 0$$

$$F_{xy} = 0$$

The solution to this problem using three equations of an infinite set is given by Timoshenko as[5]

$$N_{cr} = 24.1\pi^2 D/b^2$$

$$= 239\ D/b^2$$

Entering Table II we get for an extrapolated value of the buckling coefficient

$$K_1 = 238$$

Comparing this with the constant for the critical load we get a

$$\text{Difference} = .4\%$$

[5]Ibid., p.377.

CASE III. Simply Supported Plate Under Compressive Load Varying Linearly in the Direction of Loading.

Referring to the figure in Table III and choosing $\bar{N}$ as the common parameter we have

$$F_x = 2x/a$$

$$F_y = 0$$

$$F_{xy} = 2y/a - b/a$$

The solution to this by Libove et. al.[6] using matrix iteration methods on the matrix obtained from replacing an infinite set of equations is

$$N_{cr} = 2.92\,\pi^2 D/b^2$$

$$= 28.8\ D/b^2$$

Going to Table III we get for a coefficient

$$K_1 = 27.9$$

A comparison between the two coefficients gives a

$$\text{Difference} = 3\%$$

This difference is very much greater than for the other cases that have been considered. This may be due to the presence of shear stress whose effects are discussed in the next case.

CASE IV. Simply Supported Plate Subjected to Pure Shear.

In Table IV is shown a plate with aspect ratio $a/b = 2.5$ subjected to pure shear. For this case the common parameter is $\bar{N} = N_{xy}$ which gives the following:

$$F_x = F_y = 0$$

[6]Libove, C., S. Ferdman, and J. J. Reusch, "Elastic Buckling of a Simply Supported Plate Under a Compressive Stress that Varies Linearly in the Direction of Loading", NACA TN no. 1891, p. 18, (1949).

$$F_{xy} = 1$$

Timoshenko and Gere obtained the solution[7] to this problem by replacing an infinite set of equations with five equations and equating to zero their determinant. This gave a result of

$$N_{cr} = 6.1\pi^2 D/b^2$$
$$= 60\ D/b^2$$

Going into Table IV we get a coefficient

$$K_1 = 60$$

On the basis of the available significant figures in the published value, the solutions have a

$$\text{Difference} = 0\%$$

It will be noted that only three computed values are presented in Table IV. Originally six computed values of the buckling coefficient were used. These values indicated that the true buckling coefficient had a value lower than that obtained with the grid choice that gave the greatest number of interior nodes. The extrapolated coefficient had a value that was practically twice the value of any of the six results - which made it hard to accept as correct. An examination of the eigenvalue subroutine output revealed that the traces of the C-matrices for two grid choices were non-zero. This is contrary to what is expected for this case according to theory which will not be discussed here. For another grid choice it was found that the iteration for the eigenvalue had not converged. The three computed buckling coefficients based on these grid choices were all deemed unreliable. As a remedy to the situation, the

[7] Timoshenko, S. P. and J. M. Gere, loc. cit., p. 382.

extrapolation formula was used with only the second order terms. This required only three separate results, which were available. It will be noted that the extrapolation gives adequate accuracy for 2-significant-figure-comparison.

In Case III where shear is combined with compressive stress the effect of shear is made evident in the extrapolated value of the buckling coefficient which is closest to the computed value using the coarsest mesh. The traces and eigenvalues do not give enough information to use as a basis for discarding any data as unreliable. However, discarding the result for the coarsest mesh and extrapolating without the $H_x^2 H_y^2$ term gives a value $K_1 = 28.6$ which compares with the published value to an agreement of .7%.

CASE V. <u>Clamped Plate Under Uniform Compression</u>.

This is the same problem treated in Case I with the edge condition changed. All the stresses except N_x will be zero. S. Levy gave a solution to this problem based on an asymptotic approximation for an infinite determinant.[8] He cited other values that compared to his solution by 2-9%. He gives

$$N_{cr} = 11.659\pi^2 D/b^2$$
$$= 115.07\ D/b^2$$

The computer solution to this problem, from Table V, gives a coefficient of

$$K_1 = 115.41$$

Comparing solutions we have a

$$\text{Difference} = .3\%$$

[8] Levy, S., "Buckling of Rectangular Plates With Built-In Edges", <u>Journal of Applied Mechanics</u>, Vol. 9, pp. A171-A174, (1942).

CASE VI. Square Plate With Clamped Edges Subjected to Hydrostatic Compression and Constant Shear.

Taking the shear force as the common parameter we have

$$F_x = F_y = 1.5$$

$$F_{xy} = 1$$

The published value of the buckling coefficient is[9]

$$N_{cr} = 3.24\pi^2 D/b^2$$

$$= 32.0\ D/b^2$$

Table VI gives the computer solution to this problem as having a coefficient of

$$K_1 = 32.3$$

A comparison between coefficients gives a

Difference = .9%

CASE VII. Clamped Plate Subjected to Pure Shear.

In this problem all the stresses are zero except shear so that

$$F_x = F_y = 0$$

$$F_{xy} = 1$$

and the common parameter is the shear force itself. Interpolating from curves drawn by B. Budiansky and R. Connor in their solution to the same problem we obtain[10]

$$N_{cr} = 9.9\pi^2 D/b^2$$

$$= 97\ D/b^2$$

[9] Timoshenko, S. P. and J. M. Gere, loc. cit., p. 386.

[10] Budiansky, B. and R. W. Connor, "Buckling Stresses of Clamped Rectangular Flat Plates in Shear". NACA TN No. 1559, p. 10, (1948).

Going into Table VII we obtain for a computer solution a coefficient of

$$K_1 = 96$$

A comparison between the computer solution and the value obtained from the curves gives a

$$\text{Difference} = 1\%$$

4. Conclusions and Recommendations.

This program is capable of solving a variety of problems in buckling of rectangular plates that have simply supported or clamped edges. In the only case where an exact solution was available the extrapolated value of the buckling coefficient agreed with it to within .000016%. In other cases considered it was demonstrated that the solutions obtained using the program compare very closely to published solutions.

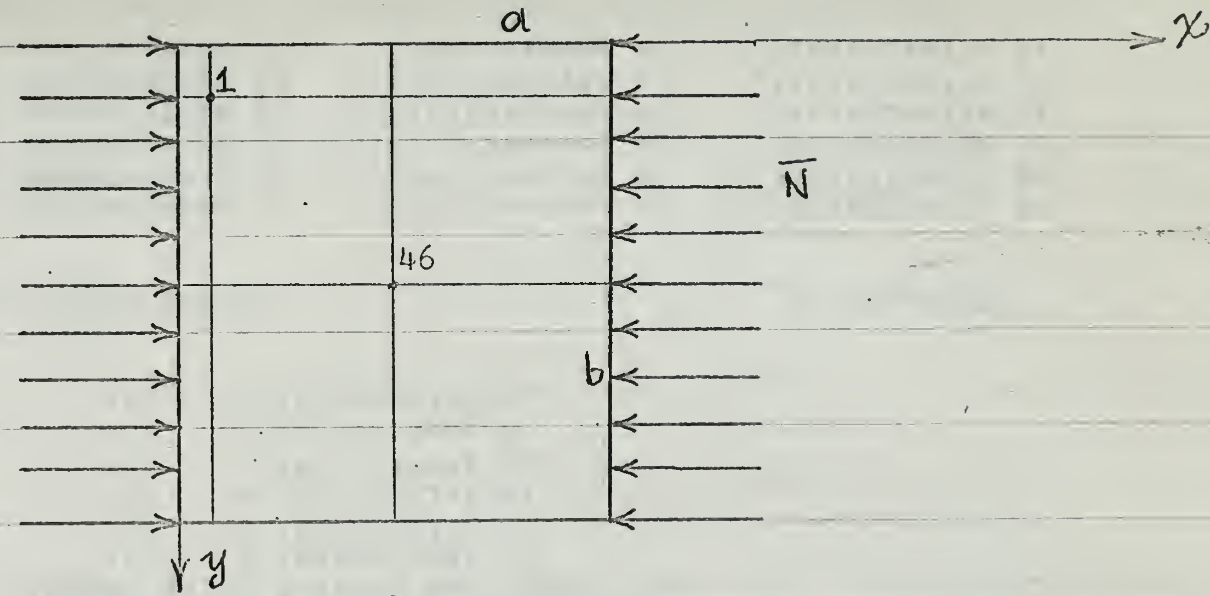
The information provided by the program output is considered adequate. The trace of the C-matrix was found useful, as exemplified in the cases solved involving pure shear. In any problem, the computed buckling coefficients give the user a rough idea of what the magnitude of the extrapolated buckling coefficient K_1 should be. This was put to use in all the cases considered where shear stress was involved. It will be profitable to evaluate the reliability of a result by studying the iterates and number of iterations which are part of the eigenvalue subroutine output. If there are 16 iterations there is a possibility that the iteration has not converged. In this event the user should examine the iterates closely.

Of the many edge conditions involved in problems of plate buckling, only two were considered. The program will prove more useful if this feature is extended to other combinations.

It is felt that the treatment of problems involving shear is inadequate. A square plate subjected to hydrostatic pressure was divided into 6 divisions in the x-direction and 12 divisions in the y-direction. It was observed that the trace of the C-matrix for such a grid choice agreed to the 9th significant figure with the trace of the C-matrix when the order of dividing the plate was reversed, i.e., 12 divisions in the x-direction and 6 divisions in the y-direction.

For a square plate subjected to hydrostatic pressure and shear the agreement is only up to the 3rd significant figure. It is evident that such a behaviour is due to shear, but the explanation has not been found.

The method of approximating the partial derivatives on the right side of the governing equation effectively required the use of a mesh twice the size of the selected mesh as far as the mixed partial derivative term is concerned. Since it is this term that is associated with shear, it might be one reason for the strange behaviour of the program when shear is involved. It is possible that the method suggested in Appendix I of using Green's Theorem to approximate the right side of the governing partial differential equation will remedy the difficulty encountered with shear. This method also makes it possible to make the C-matrix symmetric and, as a consequence, the problem of slow convergence may also be solved, since there are many eigenvalue subroutines which can handle symmetric matrices effectively.

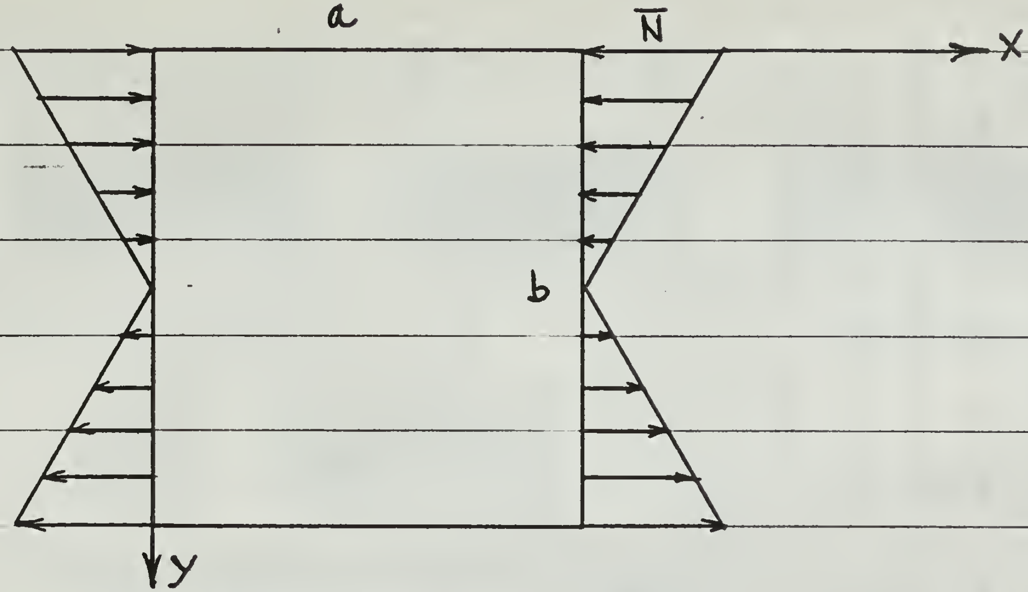


ASPECT_RATIO=.30000E+01/.40000E+01

K1= .4283681919E+02
K2= -.1096097523E+01
K3= .8628917066E-01
K4= -.1585325486E+01
K5= .4696208498E-01
K6= -.5246589561E-01

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
.600000000000E+01	.120000000000E+02	.423911629971E+02	.428368191877E+02
.800000000000E+01	.120000000000E+02	.425079995990E+02	.428368191887E+02
.900000000000E+01	.100000000000E+02	.424627132975E+02	.428368191887E+02
.100000000000E+02	.100000000000E+02	.424856639961E+02	.428368191887E+02
.600000000000E+01	.500000000000E+01	.415644206963E+02	.428368191877E+02
.100000000000E+02	.120000000000E+02	.425627771970E+02	.428368191877E+02

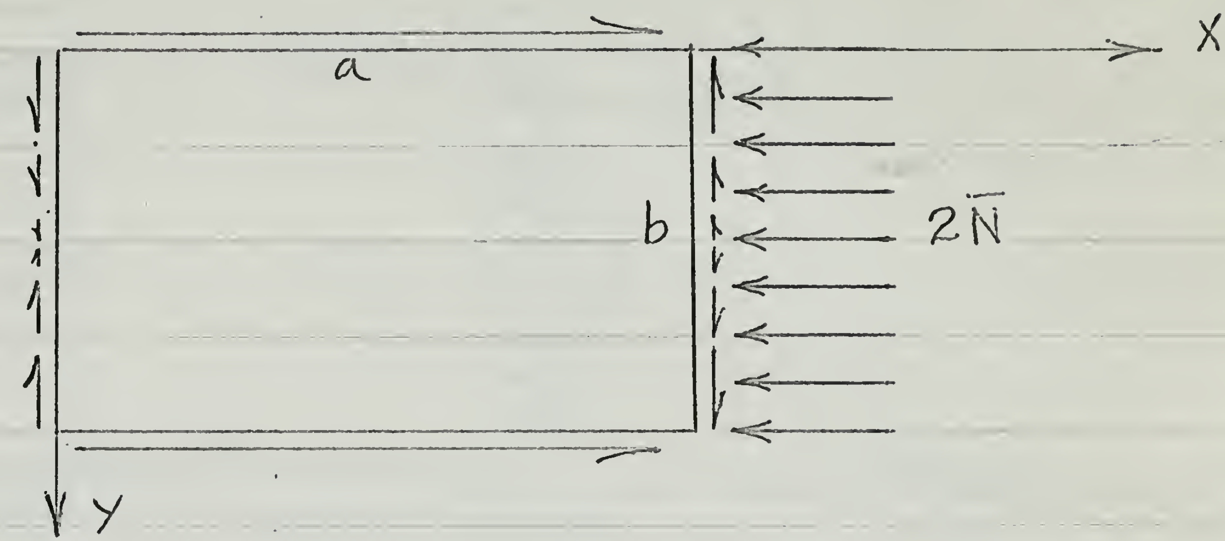
TABLE I. Simply Supported Plate Under Uniform Compression in the X-Direction




```
ASPECT RATIO=3.0000E 00/4.0000E 00
       K1=     2.3800452988E 02
       K2=     1.7691359743E 00
       K3=     6.0041035711E-01
       K4=    -4.5523333424E 01
       K5=     7.2243910606E 00
       K6=    -2.6572941972E-01
```

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
6.00000000000E 00	1.19999999998E 01	2.33507999999E 02	2.38004529872E 02
1.00000000000E 01	1.19999999998E 01	2.33197000000E 02	2.38004529872E 02
9.00000000000E 00	1.00000000000E 01	2.31104999996E 02	2.38004529878E 02
1.19999999998E 01	1.00000000000E 01	2.31015999994E 02	2.38004529878E 02
6.00000000000E 00	5.00000000000E 00	2.12265999994E 02	2.38004529878E 02
1.00000000000E 01	1.00000000000E 01	2.31065999996E 02	2.38004529878E 02

TABLE II. Simply Supported Plate Under Pure Bending Load



ASPECT RATIO=.14142E+01/.10000E+01

K1=	.2787931155E+02
K2=	.4370177732E+03
K3=	-.6965436232E+05
K4=	-.4040858000E+03
K5=	-.6731012556E+05
K6=	.1354971921E+06

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
.800000000000E+01	.600000000000E+01	.279719383274E+02	.278793115467E+02
.900000000000E+01	.700000000000E+01	.282010700461E+02	.278793115504E+02
.100000000000E+02	.800000000000E+01	.283538178816E+02	.278793115485E+02
.110000000000E+02	.900000000000E+01	.284746263763E+02	.278793115467E+02
.120000000000E+02	.900000000000E+01	.284981513699E+02	.278793115458E+02
.120000000000E+02	.100000000000E+02	.285597742680E+02	.278793115495E+02

TIME, 0 MINUTES AND 21 SECONDS

TABLE III. Simply Supported Plate Under Uniform Compression Which Varies Linearly in the Direction of Loading.

ASPECT RATIO=.25000E+01/.10000E+01

K1=	.5983924994E+02
K2=	.1084438640E+02
K3=	.3264925589E+03

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
.800000000000E+01	.600000000000E+01	.699675097931E+02	.598392499350E+02
.900000000000E+01	.700000000000E+01	.673391215950E+02	.598392499387E+02
.110000000000E+02	.900000000000E+01	.644301661961E+02	.598392499387E+02

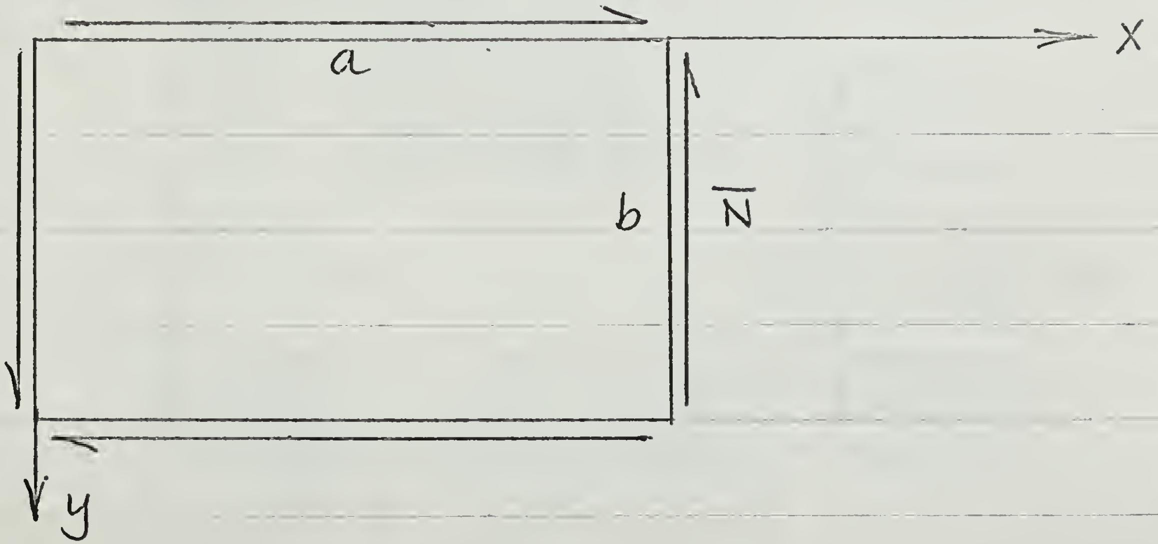
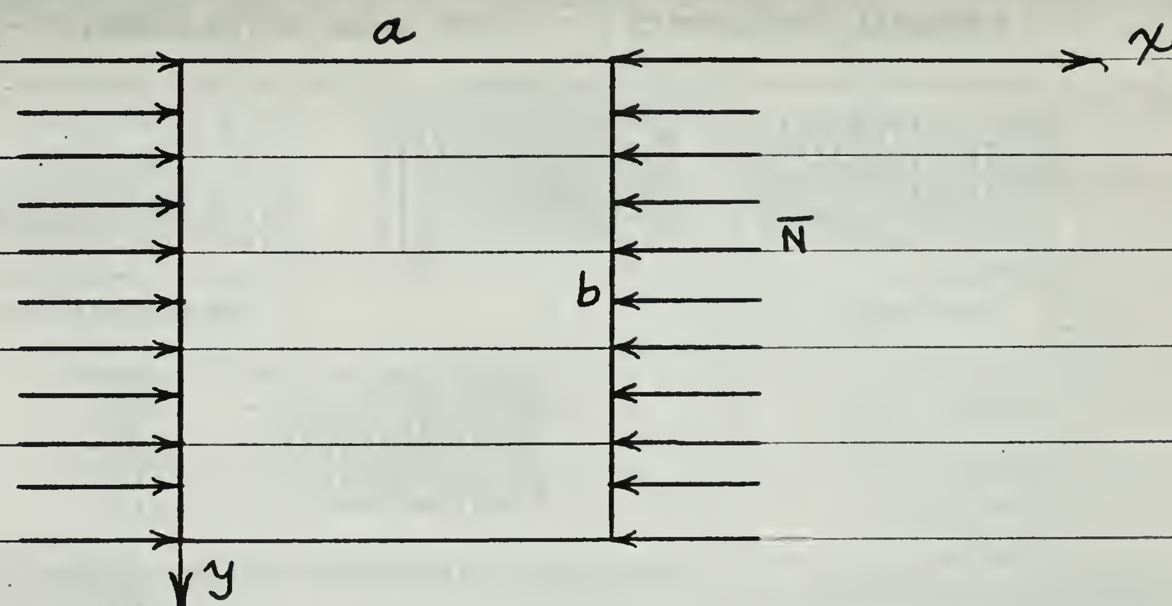


TABLE IV. Simply Supported Plate Subjected to Pure Shear

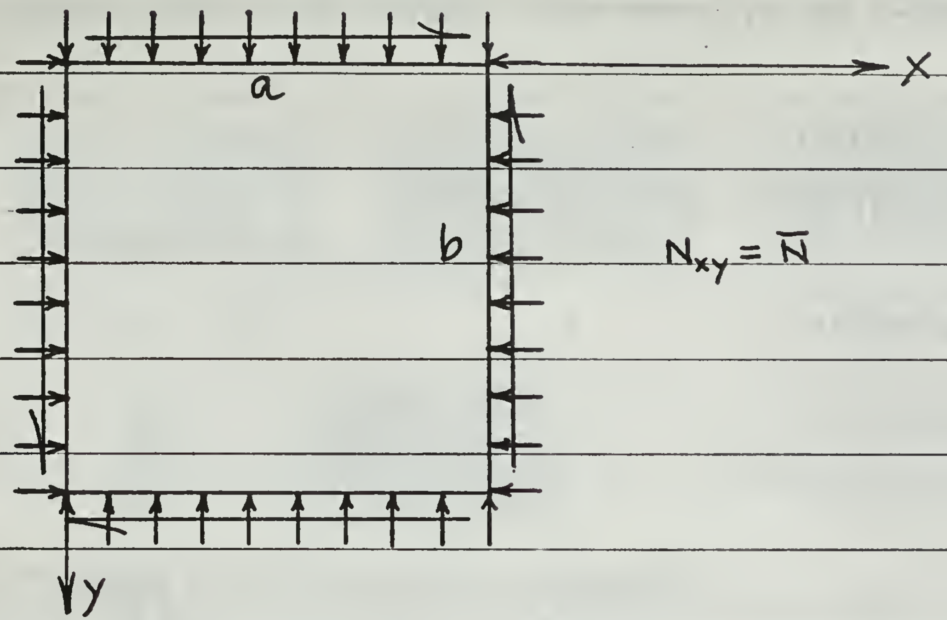


ASPECT RATIO=.30000E+01/.40000E+01

K1=	.1154115245E+03
K2=	-.2517933263E+02
K3=	.1040329272E+02
K4=	-.2148144138E+02
K5=	.2433492588E+01
K6=	.2472791760E+02

DIVISIONS ALCNG X	DIVISIONS ALONG Y	K	K,CORRECTED
.600000000000E+01	.120000000000E+02	.108096999995E+03	.115411524460E+03
.800000000000E+01	.120000000000E+02	.110105999997E+03	.115411524460E+03
.100000000000E+02	.100000000000E+02	.110210999997E+03	.115411524462E+03
.120000000000E+02	.100000000000E+02	.110750999998E+03	.115411524462E+03
.600000000000E+01	.500000000000E+01	.100971999995E+03	.115411524460E+03
.800000000000E+01	.900000000000E+01	.108614999996E+03	.115411524460E+03

TABLE V. Clamped Plate Under Uniform Compression in the X-Direction

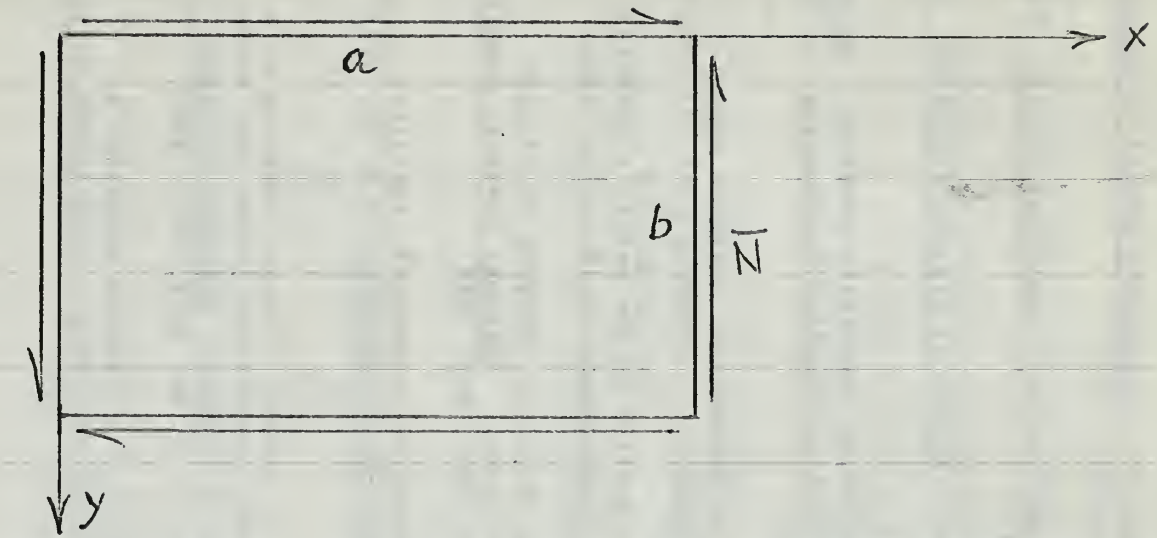


ASPECT RATIO=.10000E+01/.10000E+01

K1=	.3234480971E+02
K2=	-.3718934603E+02
K3=	.9341002566E+03
K4=	-.1809689921E+02
K5=	-.3003858573E+03
K6=	-.2053048955E+04

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
.120000000000E+02	.600000000000E+01	.310010910984E+02	.323448097054E+02
.600000000000E+01	.120000000000E+02	.314963338976E+02	.323448097063E+02
.110000000000E+02	.100000000000E+02	.317205789983E+02	.323448097054E+02
.100000000000E+02	.100000000000E+02	.316500137984E+02	.323448097063E+02
.800000000000E+01	.800000000000E+01	.311344448989E+02	.323448097063E+02
.700000000000E+01	.700000000000E+01	.306253758986E+02	.323448097063E+02

TABLE VI. Clamped Plate Under Shear and Hydrostatic Pressure



ASPECT RATIO=.25000E+01/.10000E+01

K1= .9651404821E+02
K2= .1044229604E+03
K3= .1196966901E+03

DIVISIONS ALONG X	DIVISIONS ALONG Y	K	K,CORRECTED
.800000000000E+01	.600000000000E+01	.110036510995E+03	.965140482038E+02
.900000000000E+01	.700000000000E+01	.107014164995E+03	.965140482076E+02
.130000000000E+02	.900000000000E+01	.101853580996E+03	.965140482076E+02

TABLE VII. Clamped Plate Subjected to Pure Shear

Bibliography

1. Kantorovich, L. V. and V. I. Krylov Approximate Method of Higher Analysis, 3rd ed. Interscience, 1958.

2. Scarborough, J. B. Numerical Mathematical Analysis, 5th ed. The Johns Hopkins Press, 1962.

3. Timoshenko, S. P. and J. M. Gere Theory of Elastic Stability, 2nd ed. McGraw-Hill, 1961.

4. Griffin, D. S. and R. S. Varga Plane elasticity problems. Journal of the Society for Industrial and Applied Mathematics, v. 11, December, 1963: 1046-1062.

5. Levy, S. Buckling of rectangular plates with built-in edges. Journal of Applied Mechanics, v. 9, December 1942: A171-A174.

6. Parlett, B. Laguerre's method applied to the matrix eigenvalue problem. Mathematics of Computation, v. 18, July, 1964.

7. Budiansky, B. and R. W. Connor Buckling stresses of clamped rectangular flat plates in shear. National Advisory Committee for Aeronautics, T. N. No. 1559, May, 1948.

8. Gerard, G. and H. Becker Handbook of structural stability. National Advisory Committee for Aeronautics, T.N. No. 3781. July, 1957.

9. Libove, C., S. Ferdman and J. J. Reusch, Elastic buckling of a simply supported plate under compressive stress that varies linearly in the direction of loading. National Advisory Committee for Aeronautics, T. N. No. 1891, June, 1949.

10. Yardley, J. Applications of finite difference equations to buckling problems of rectangular plates. Unpublished master's thesis, Washington University, 1948.

APPENDIX I

MATHEMATICAL DEVELOPMENTS

I- Finite Difference Approximations.

The partial differential equation to be approximate is

$$\nabla^4 w = \frac{1}{D}\left(N_x \frac{\partial^2 w}{\partial x^2} + 2 N_{xy} \frac{\partial^2 w}{\partial x \partial y} + N_y \frac{\partial^2 w}{\partial y^2}\right) \tag{I-1}$$

where

$$\nabla^4 w = \frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}$$

I-A. Simply Supported Case.

The boundary conditions at all edges of the plate are

$$w = 0 \tag{I-2}$$

$$\nabla^2 w = 0 \tag{I-3}$$

where we define ∇^2 as the Laplacian operator which is

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$ in two dimensions.

Since it is known that the values of w = 0 at the boundaries by virtue of B.C. (I-2) and, since we need only as many equations as there are unknowns, only equations for interior points need be written.

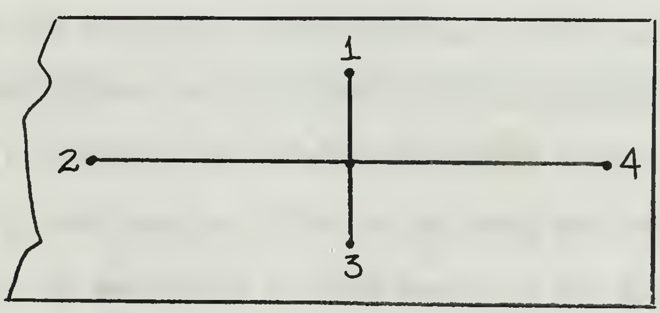


Fig. I-1

The left side of Eqn. I-1 may be approximated easily once $\nabla^2 w$ is obtained. This is done with reference to Fig. I-1. Let the numbered points be the mesh nodes when the plate is divided into meshes with the mesh sides of length H_x and H_y. At point #0 using the conventional method of approximating the partial derivative we have

$$\frac{\partial^2 w}{\partial x^2} = \frac{w_4 + w_2 - 2w_0}{H_x^2} \tag{I-4}$$

$$\frac{\partial^2 w}{\partial^2 y^2} = \frac{w_1 + w_3 - 2w_0}{H_y^2} \tag{I-5}$$

$$\nabla^2 w_0 = \frac{1}{H_x H_y}\left(\left(w_1 + w_3\right)\frac{H_x}{H_y} + \left(w_2 + w_4\right)\frac{H_y}{H_x} - 2w_0\left(\frac{H_x}{H_y} + \frac{H_y}{H_x}\right)\right) \tag{I-6}$$

An examination of Eqn. I-6 shows that for a given point on the plate, four other points will be involved in writing the equation for $\nabla^2 w$. The deflection w_i at each point will have the following coefficients when $1/(H_x H_y)$ remains factored out:

$$a_1 = a_3 = \frac{H_x}{H_y}$$

$$a_2 = a_4 = \frac{H_y}{H_x}$$

$$a_0 = - a_1 - a_2 - a_3 - a_4 \tag{I-7}$$

When one of the points involved (other than point #0) lies on a boundary, a slight modification in obtaining the coefficients must be performed. If in Fig. I-1 point #4 lies on the boundary, its contribution in Eqn. I-7 for the point #0 is still included, but a_4 will not appear in the A-matrix and w_4 is not used in assembling the deflection vector. We have seen that

$$\nabla^2 w_0 = \frac{1}{H_x H_y}\left(a_1 w_1 + a_2 w_2 + a_3 w_3 + a_4 w_4 - a_0 w_0\right) \tag{I-8}$$

In general, for n mesh points, Eqn. I-8 may be written as

$$\nabla^2 w = Aw \tag{I-9}$$

where

A = an n x n symmetric matrix

w = a column vector of lateral deflections at interior nodes of the plate.

For a square mesh the deflection coefficients will be

$$a_i = 1 \qquad i = 1,2,3,4$$

$$a_o = -4$$

Matrix A is easily assembled if we use the five-point cross for the coefficients as illustrated in Fig. I-2. The number on each node represents the coefficient of w at that node.

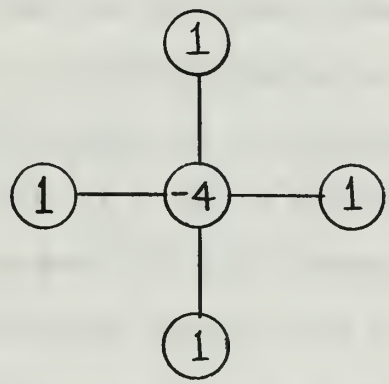


Fig. I-2

Having obtained $\nabla^2 w$ we are now ready to approximate $\nabla^4 w$. To get $\nabla^4 w$ we simply operate on Eqn. I-9 so that

$$\nabla^4 w = \nabla^2 (\nabla^2 w) = A(Aw) = A^2 w \tag{I-10}$$

It should be noted that $\nabla^2 w$ was formulated under the condition that w = 0 at the boundaries and $\nabla^4 w$ was formulated under the condition that $\nabla^2 w$ = 0 at the boundaries also.

It has been shown that only $\nabla^2 w$ need be developed to obtain $\nabla^4 w$. The scheme used to write down the matrix for $\nabla^2 w$ will be discussed next.

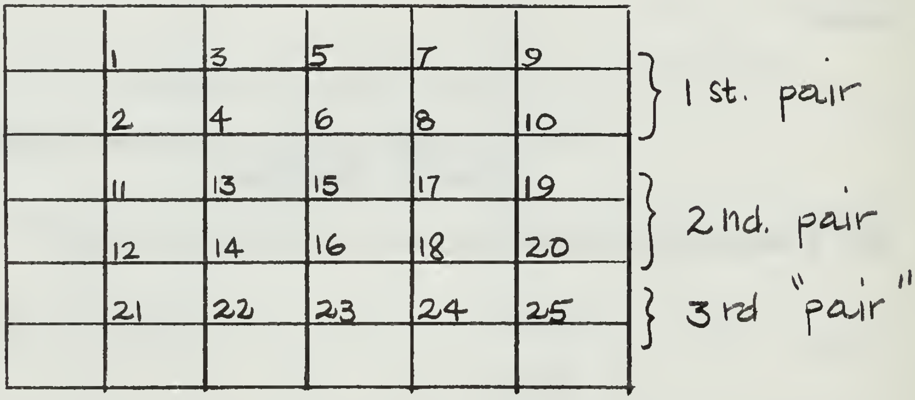


Fig. I-3

Let the mesh nodes be numbered in the manner shown in Fig. I-3[11], where two adjacent horizontal mesh lines are taken as a pair and each node on the lines is numbered alternating between the two lines. If we write down the approximations to Eqn. I-4 for each point on the plate in the sequence that they are numbered in Fig. I-3, using a square mesh a coefficient matrix such as shown in Fig. I-7 results. This is the matrix that must be multiplied by itself to get the coefficient matrix A^2 for $\nabla^4 w$. If we write the equation for a given point i the deflection coefficient will lie in the ith row and in the ith column. The other points involved in the equation for point i will have a deflection coefficient lying in the ith row and in columns bearing their respective numbers. Thus it is easy to locate the deflection coefficient of any point. In

[11] The scheme is based on the work of Griffin, D. S. and R. S. Varga in their paper "Numerical Solution of Plane Elasticity Problems", <u>J. Industrial and Applied Math.</u>, Vol. 11, pp. 1046-1061, (1963).

Fig. I-7 which was assembled using a square mesh, the deflection at point 4 will have a coefficient of -4 in row 4, column 4. Points 2,3,6, and 15 will have deflection coefficients equal to 1 in the same row and lying in columns 2,3,6, and 15, respectively.

Squaring matrix A we will get the matrix for $\nabla^4 w$ which is illustrated in Fig. I-8. It is now simple to check this matrix. If we follow the same procedure of locating the elements in any row it can be verified that the matrix A^2 which was assembled for a square mesh agrees with the matrix formed using the 13-point star to approximate $\nabla^4 w$ for a square mesh. This star is illustrated in Fig. I-4 which shows the deflection coefficient for each point.

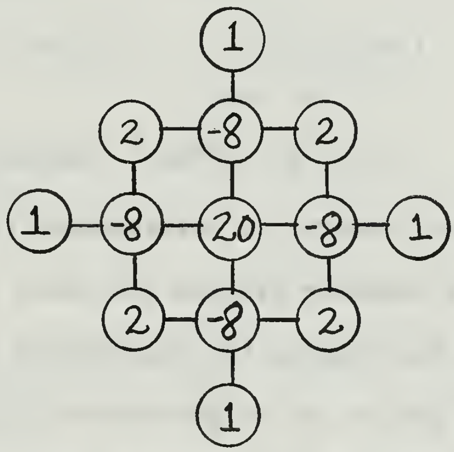


Fig. I-4

This completes the approximation to the left side of the governing partial differential equation.

Approximations of the Right Side of Eqn. I-1.

The first attempt to approximate the right side of Eqn. I-1 was to apply Green's Theorem to the integrals

$$\frac{1}{D}\left(\iint N_x \frac{\partial^2 w}{\partial x^2} dx dy + 2\iint N_{xy} \frac{\partial^2 w}{\partial x \partial y} dx dy + \iint N_y \frac{\partial^2 w}{\partial y^2} dx dy\right)$$

Applying the theorem will result in a symmetric coefficient matrix B. Since A and A^2 are symmetric, we may then deal only with symmetric matrices. The C-matrix can be made symmetric also by the following steps.

$$A^2 w = kBw$$

$$A(Aw) = kBA^{-1}(Aw)$$

$$Aw' = kBA^{-1}w'$$

$$w' = kA^{-1}BA^{-1}w'$$

$$w' = kCw'$$

where

$$w' = Aw$$

and $C = A^{-1}BA^{-1}$ which is symmetric if B is symmetric. A symmetric C-matrix is advantageous because less storage is required by the digital computer program and there is a wide choice of dependable subroutines for finding the eigenvalues of a symmetric matrix. However, a symmetric B-matrix can be obtained only with the use of stresses at intermediate points between nodes. The resulting matrix will be more intricate and programming more elaborate since a problem in "bookkeeping" arises from the use of stresses at points other than at the point under consideration. For these reasons a direct approximation to each term in the right side of Eqn. I-1 was taken using the classical method of approximating the slope at a given point.

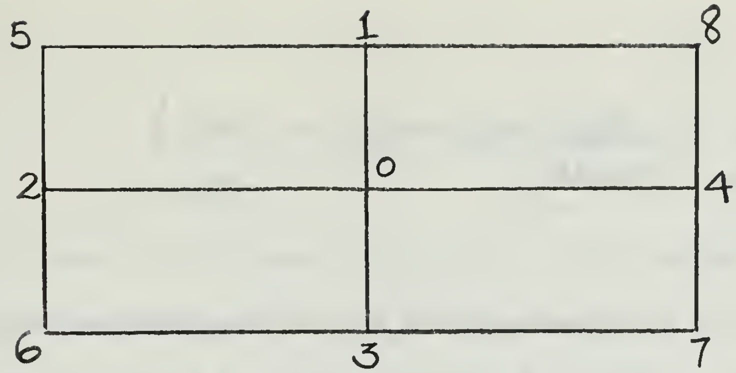


Fig. I-5

Using Fig. I-5 and remembering that H_x and H_y are constant throughout the entire plate, the right side of Eqn. I-7 may be approximated term by term in the following manner:

$$\frac{\partial^2 w}{\partial x^2} = \frac{1}{H_x}\left(\frac{w_4 - w_0}{H_x} - \frac{w_0 - w_2}{H_x}\right)$$

$$N_x \frac{\partial^2 w}{\partial x^2} = \frac{N_x}{H_x^2}\left(w_4 + w_2 - 2\,w_0\right) \qquad \text{(I-11)}$$

similarly

$$N_y \frac{\partial^2 w}{\partial y^2} = \frac{N_y}{H_y^2}\left(w_1 + w_3 - 2\,w_0\right) \qquad \text{(I-12)}$$

For the mixed partial derivative

$$\left(\frac{\partial w}{\partial y}\right)_4 = \frac{1}{2H_y}\left(w_8 - w_7\right)$$

$$\left(\frac{\partial w}{\partial y}\right)_2 = \frac{1}{2H_y}\left(w_5 - w_6\right)$$

$$\left(\frac{\partial^2 w}{\partial x \partial y}\right)_0 = \frac{1}{2H_x}\left(\left(\frac{\partial w}{\partial y}\right)_4 - \left(\frac{\partial w}{\partial y}\right)_2\right)$$

$$N_{xy}\frac{\partial^2 w}{\partial x \partial y} = \frac{N_{xy}}{4H_xH_y}\left(w_8 + w_6 - w_7 - w_5\right) \qquad \text{(I-13)}$$

For the purpose of programming, $1/(H_xH_y)$ was factored out as in the development for A. Combining Eqns. I-11, I-12 and I-13 and remembering that N_x, N_y, and N_{xy} are expressed in terms of $\overline{N}$ and F_x, F_y, and F_{xy} we obtain

$$\text{Right side} = \frac{\overline{N}}{DH_xH_y}\left(F_x\left(w_4 + w_2 - 2w_0\right)\frac{H_y}{H_x} + F_y\left(w_1 + w_3 - 2w_0\right)\frac{H_x}{H_y} + \frac{F_{xy}}{2}\left(w_8 + w_6 - w_7 - w_5\right)\right)$$

The coefficient of w at each node involved may be summarized into

$$b_0 = -2(F_xH_y/H_x + F_yH_x/H_y)$$

$$b_1 = b_3 = F_yH_x/H_y$$

$$b_2 = b_4 = F_xH_y/H_x$$

$$b_5 = b_7 = -F_{xy}/2$$

$$b_6 = b_8 = F_{xy}/2$$

It has been shown that the right side of Eqn. I-1 may be written in the form

$$\text{Right side} = \frac{1}{DH_xH_y}\Big(b_1w_1 + b_2w_2 + b_3w_3 + b_4w_4 + b_5w_5 + b_6w_6 + b_7w_7 + b_8w_8 - b_0w_0\Big)$$

In general, for n interior mesh nodes we will have the matrix equation

Right side = kBw

where

$$B = \text{an } n \times n \text{ coefficient matrix}$$

$$k = \overline{N}/D$$

An example of matrix B is shown in Fig. I-9. This corresponds to the plate illustrated in Fig. I-8. The loading is

$$F_x = F_y = -1$$

$$F_{xy} = 2$$

The common parameter in this case $\overline{N} = -N_x = -N_y$. This completes the approximation to the right side of Eqn I-1.

It will be noted that the pairing of mesh lines in Fig. I-3 depends on the divisions only in the y-direction. It is readily seen that the matrices will be different for an odd number than for an even number of divisions along the y-direction. The matrices for a plate divided into an even number of divisions along the y-direction have been illustrated in Figs. I-7, I-8, and I-9. Figs. I-10, I-11, and I-12 illustrate the respective matrices for an odd number of divisions in the y-direction.

Symmetrical Cases

There are cases where the load and deflection are symmetrical with respect to the geometric axes of symmetry of the plate. When this is known (or shown by eigenvectors obtained using the whole plate) we may choose to use only one quadrant of the plate. This allows use of a finer mesh without a corresponding increase in computer storage requirements. It is the intention here that only the case when the midpoint of the plate has the maximum deflection will be handled by the program. However, where the buckled surface assumes a complete cycle of a sine curve,

say in the x-direction, then one-half of the plate may be considered and if it satisfies the conditions of symmetry as described previously, a quarter of this may be chosen for the program to handle.

The procedure for assembling the matrices for a quadrant of a plate is the same as in the preceding section with the following modification. The whole plate is always divided so that the horizontal and vertical axes of symmetry are taken as mesh lines. Going back to Fig. I-5, suppose that points 0, 1, and 3 are lying on the line of symmetry and points 2, 5, and 6 are in the quadrant being analyzed. The coefficients a_1, a_3 and a_0 are computed as before but a_2, a_5, and a_6 are now doubled since the following conditions obtain:

$$a_5 = a_8 \qquad w_5 = w_8$$

$$a_2 = a_4 \qquad w_2 = w_4$$

$$a_6 = a_7 \qquad w_6 = w_7$$

Points 8, 4, and 7 are not used in assembling the matrix. Squaring A does not give a symmetric matrix. To make A^2 symmetric it has to be premultiplied by a diagonal matrix whose elements are proportional to the areas of the mesh regions associated with the corresponding matrix rows. This matrix may be designated as A_1^2 and is used for obtaining matrix C for the quarter plate.

The procedure described also holds equally well for the right hand side of the governing equation. B_1 is obtained after multiplying matrix B generated for the quadrant of the plate with the diagonal matrix used in obtaining A_1^2.

I-B. Clamped Edges.

The boundary conditions at all edges are

$$w = 0 \tag{I-14}$$

$$\frac{\partial w}{\partial x} = 0 \tag{I-15}$$

$$\frac{\partial w}{\partial y} = 0 \tag{I-16}$$

Approximation of the Left Side of Eqn. I-1.

Returning to the developments in the case of simply supported edges, it can be demonstrated that the development of A still holds. Since the first boundary condition for both cases is the same and since it is the only boundary condition used to assemble the coefficient matrix for $\nabla^2 w$, the matrices will be identical for both edge conditions. In assembling the coefficient matrix for $\nabla^4 w$ it will be necessary to take into account that $\nabla^2 w$ is nonzero on the boundaries. In Fig. I-6 we have

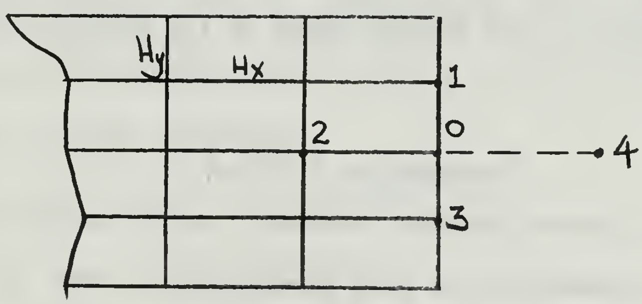


Fig. I-6

$$\nabla^2 w_0 = \frac{1}{H_x H_y}\left((w_1 - w_0)\frac{H_x}{H_y} + (w_2 - w_0)\frac{H_y}{H_x} + (w_3 - w_0)\frac{H_x}{H_y} + (w_4 - w_0)\frac{H_y}{H_x}\right) \tag{I-17}$$

Applying the conditions that at the right edge

$$w_0 = w_1 = w_3 = 0$$

and

$$w_2 = w_4 \text{ by virtue of B.C. I-15}$$

$$\nabla^2 w_0 = \frac{2w_2}{H_x H_y}\,\frac{H_y}{H_x} = \frac{2w_2}{H_x^2} \tag{I-18}$$

When we take $\nabla^4 w$ by operating on Eqn. I-7 there will be a term in $\nabla^2 w_0$ which is nonzero (unlike the case of the simply supported plate). The coefficient of the term will be the correction necessary to convert the matrix for simply supported plate into that of a clamped plate. This correction will be of the form

$$a_{correction} = \frac{2}{H_x^2}\left(\frac{1}{H_x H_y}\cdot\frac{H_y}{H_x}\right)$$
$$= 2/H_x^4$$

Excluding the "corner" points, the correction above is good for all points adjacent to the right and left edges of the plate. For points adjacent to the top and bottom edges of the plate the correction will be

$$a_{correction} = 2/H_y^4$$

For the "corner" points there will be a contribution from points lying on the boundaries in both directions so that the correction term will be a combination of the corrections above or

$$a_{correction} = 2(1/H_x^4 + 1/H_y^4)$$

In terms of the matrices

$$A^2\text{, clamped} = A^2\text{, simply supported} + A_{correction}$$

where

$$A_{correction} = n \times n \text{ diagonal matrix of correction terms.}$$

Fig. I-13 illustrates the coefficient matrix for A^2 using even numbers of divisions along the width of a clamped plate.

Approximation of the Right Side of Eqn. I-1.

It has been noted that the simply supported plate and the clamped plate had zero deflections along the boundaries. This leads to the conclusion that the assembly of matrix B for a clamped plate will be exactly the same as for the simply supported plate.

Estimate of the Error.

The finite difference method of solving a partial differential equation carries with it an inherent error resulting from the replacement of an infinitesimal quantity with one that is finite. To get a satisfactory answer the use of a considerable number of interior points will be necessary. When this is not a possibility one may use some form of extrapolation formula to get a good approximation to the real solution.

There were two schemes tried to evaluate the error resulting from the finite difference approximations to Eqn. I-1. The first was to assume that the error took the form

$$\epsilon = C_1 H_x^{n_x} + C_2 H_y^{n_y}$$

where ϵ = the error

C_1, C_2, n_x, n_y = constants

The calculated and extrapolated values of the buckling coefficient then have the relationship

$$K_c = K_1 + C_1 H_x^{n_x} + C_2 H_y^{n_y}$$

where

K_c = computed buckling coefficient

K_1 = the extrapolated buckling coefficient

For five computed values of K_c the extrapolated value K_1 of the buckling coefficient can be found. This has been done and, while the results were not very far from those obtained analytically, the method which will be described below was found to give better results.

It has been shown[12] that the error in approximating $\nabla^4 w$ with finite differences using a square mesh can be expressed as

$$\epsilon = K_2 H^2 + K_3 H^4 + K_4 H^6 + K_5 H^8 + \cdot \cdot \cdot \cdot \cdot + K_n H^{2n}$$

where

$K_2, K_3, \ldots . K_n$ = constants

H = length and width of the mesh used.

It has been shown[13] also that the same form of expression holds true for $\nabla^2 w$. The difference between the two will lie only in the constants. While such an expression applies only to $\nabla^4 w$ and $\nabla^2 w$ it is reasonable to expect that the error for the governing partial differential equation will vary with even powers of the mesh sides because of its form which is practically a combination of $\nabla^2 w$ and $\nabla^4 w$. It will be assumed that the error will vary according to:

$$\epsilon = K_2 H_x^2 + K_3 H_x^4 + K_4 H_y^2 + K_5 H_y^4 + K_6 H_x^2 H_y^2$$

The relationship between the computed buckling coefficient and the extrapolated value will again take a form similar to Eqn. I-19. For six computed values of K_c the extrapolated value K_1 of the buckling coefficient can be determined.

[12] Kantorovich, L. V. and V. I. Krylov, Approximate Methods of Higher Analysis (New York: Interscience Publishers, 1958), p. 196.

[13] Scarborough, J. B., Numerical Mathematical Analysis, (Baltimore: The Johns Hopkins Press, 1962), p. 399.

-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.								
1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
.	1.	1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.								
.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.								
.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.								
.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	1.	.	-4.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.
.	1.	.	.	.	.	.	.	.	.	.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	1.	-4.	.	1.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.
.	.	.	1.	.	.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	.	1.	.	.	.	.
.	.	.	.	.	1.	.	.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.
.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	1.	.	.
.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	1.	.
.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	1.	.	-4.	1.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	.	.	.	.	1.
.	.	.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	-4.	1.	.	.	.	.
										.	.	.	.	.	1.	.	.	.	.	.	.	.	.	1.	-4.	1.	.	.	.
										.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	-4.	1.	.	.
										.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	-4.	1.	.
										.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	1.	-4.	1.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	1.	-4.

	1	3	5	7	9	11
	2	4	6	8	10	12
	13	15	17	19	21	23
	14	16	18	20	22	24
	25	26	27	28	29	30

Fig.I-7

Matrix A for a Simply Supported Plate With Even Number of Divisions Along Its Width.

18.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.								
-8.	19.	2.	-8.	.	1.	.	.	.	.	.	.	-8.	1.	2.	.	.	.	.	.	.	.								
-8.	2.	19.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.								
2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.								
1.	.	-8.	2.	19.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.								
.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.								
.	.	1.	.	-8.	2.	19.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.								
.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.	.	.	.	.
.	.	.	.	1.	.	-8.	2.	19.	-8.	-8.	2.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.	.	.
.	.	.	.	.	.	1.	.	-8.	2.	18.	-8.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	1.	2.	-8.	-8.	19.	.	.	.	.	.	.	.	.	2.	.	-8.	1.	.	.	.	.	.	.
1.	-8.	.	2.	.	.	.	.	.	.	.	.	19.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.
.	1.	.	.	.	.	.	.	.	.	.	.	-8.	19.	2.	-8.	.	1.	.	.	.	.	.	.	-8.	2.	.	.	.	.
.	2.	1.	-8.	.	2.	.	.	.	.	.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	.	1.	.	.	.	.
.	.	.	1.	.	.	.	.	.	.	.	.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	-8.	2.	.	.	.
.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.
.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	2.	-8.	2.	.	.
.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	1.	.	.
.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	-8.	2.	.
.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	.	.	.	.	1.	.
.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	2.	-8.	2.
.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	.	.	.	1.	.	-8.	2.	19.	-8.	.	.	.	.	.	1.
.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	19.	.	.	.	.	2.	-8.
.	.	.	.	.	.	.	.	.	.	.	.	1.	-8.	.	2.	.	.	.	.	.	.	.	.	18.	-8.	1.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	.	.	-8.	19.	-8.	1.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	-8.	19.	-8.	1.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	1.	-8.	19.	-8.	1.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	1.	-8.	19.	-8.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	1.	-8.	18.

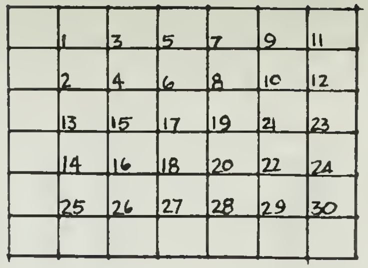


Fig.I-8

A^2-matrix for a Simply Supported Plate With Even Number of Divisions Along Its Width.

TIME, 1 MINUTE AND 0 SECONDS

-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
1.	-4.	1.	1.	.	.	.	.	.	.	.	.	1.	.	-1.	.	.	.	.	.	.	.								
1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.								
.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.								
.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	1.	.	-1.	.	.	.								
.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.								
.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	1.	1.	-4.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	.	.	.	.	.	.	.	.	1.	.	1.	.	.	.	.	.	.	.
.	1.	.	1.	.	.	.	.	.	.	.	.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	1.	-4.	1.	1.	.	.	.	.	.	.	.	.	1.	-1.	.	.	.	.
.	-1.	.	1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	1.	-1.	.	.	.
.	.	.	-1.	.	1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	.	1.	1.	-1.	.	.
.	.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	1.	-1.	.
.	.	.	.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	1.	1.	-1.
.	.	.	.	.	.	.	.	.	-1.	.	1.	.	.	.	.	.	.	.	.	1.	1.	-4.	1.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	.	.	.	.	1.	1.
.											.	.	1.	.	1.	.	.	.	.	.	.	.	.	-4.	1.	.	.	.	.
.											.	.	-1.	.	1.	.	1.	.	.	.	.	.	.	1.	-4.	1.	.	.	.
.											.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	.	1.	-4.	1.	.	.
.											.	.	.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	1.	-4.	1.	.
.											.	.	.	.	.	.	.	.	-1.	.	1.	.	1.	.	.	.	1.	-4.	1.
—						.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	.	1.	.	.	.	.	1.	-4.

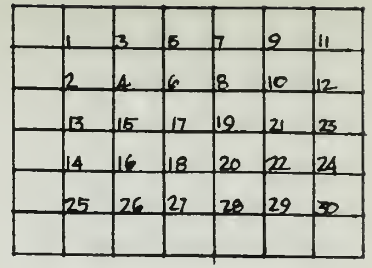


Fig.I-9

Matrix B for a Simply Supported Plate With Even Number of Divisions Along Its Width.

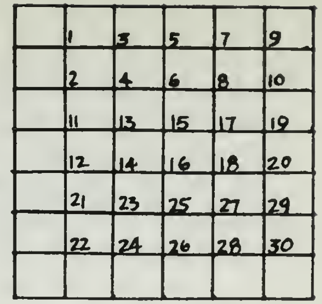
TIME, 1 MINUTE AND 36 SECONDS

-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	.	.							
1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.							
.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
.	.	.	1.	1.	4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.							
.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.							
.	.	.	.	.	.	1.	.	-4.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	1.	1.	-4.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	.
.	1.	.	.	.	.	.	.	.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.
.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.
.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.	.	.
.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.	.	.	1.	.	.	.
.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	.	.	.	.	.	.	.	1.	.
.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	-4.	1.	1.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	-4.	.	1.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.	.	.
										.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.	.	.
										.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.	.	.
										.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.	1.	.
										.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.	.	1.
										.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	1.	.	-4.	1.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	1.	-4.

	1	3	5	7	9
	2	4	6	8	10
	11	13	15	17	19
	12	14	16	18	20
	21	23	25	27	29
	22	24	26	28	30

Fig.I-10

Matrix A for a Simply Supported Plate With Odd Number of Divisions Along Its Width.



18.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	.	.							
-8.	19.	2.	-8.	.	1.	.	.	.	.	-8.	1.	2.	.	.	.	.	.	.	.	.	.	.							
-8.	2.	19.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.							
2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.	.	.	.							
1.	.	-8.	2.	19.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.							
.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.	.							
.	.	1.	.	-8.	2.	19.	-8.	-8.	2.	.	.	.	.	.	.	1.	.	.	.	.	.	.							
.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.							
.	.	.	.	1.	.	-8.	2.	18.	-8.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	1.	2.	-8.	-8.	19.	.	.	.	.	.	.	2.	.	-8.	1.	.	.	.	.	.	.	.	.	.	.
1.	-8.	.	2.	.	.	.	.	.	.	19.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.
.	1.	.	.	.	.	.	.	.	.	-8.	19.	2.	-8.	.	1.	.	.	.	.	-8.	1.	2.	.	.	.	.	.	.	.
.	2.	1.	-8.	.	2.	.	.	.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.	.	.
.	.	.	1.	.	.	.	.	.	.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.
.	.	.	2.	1.	-8.	.	2.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.	.	.
.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	.	-8.	1.	2.	.	.	.
.	.	.	.	.	2.	1.	-8.	.	2.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	.	.	.	.	.	.	1.	.	.	.
.	.	.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	.	2.	.	-8.	1.	2.	.
.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	.	1.	.	-8.	2.	19.	-8.	.	.	.	.	.	.	.	.	1.	.
.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	19.	.	.	.	.	.	.	2.	.	-8.	1.
.	.	.	.	.	.	.	.	.	.	1.	-8.	.	2.	.	.	.	.	.	.	19.	-8.	-8.	2.	1.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	-8.	18.	2.	-8.	.	1.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	2.	-8.	-8.	19.	2.	-8.	.	1.	.	.
										.	.	.	2.	1.	-8.	.	2.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.
										.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	19.	2.	-8.	.	1.
										.	.	.	.	.	2.	1.	-8.	.	2.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.
										.	.	.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	19.	2.	-8.
										.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	.	1.	.	-8.	2.	19.	-8.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	1.	2.	-8.	-8.	18.

Fig.I-11

A^2-matrix for a Simply Supported Plate With Odd Number of Divisions Along Its Width.

-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	-1.	.	.	.	.	.	.	.	.	.	.							
1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.	.	.	.							
.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.							
.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.	.							
.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.							
.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.							
.	.	.	.	.	.	1.	1.	-4.	1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	-1.	1.	1.	-4.	.	.	.	.	.	.	1.	.	1.	.	.	.	.	.	.	.	.	.	.	.
.	1.	.	1.	.	.	.	.	.	.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	1.	-4.	1.	1.	.	.	.	.	.	.	1.	.	-1.	.	.	.	.	.	.	.
.	-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.	.	.	.	.
.	.	.	-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.	.	.
.	.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.	1.	.	1.	.	-1.	.
.	.	.	.	.	.	.	-1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	.	.	.	.	.	.	1.	.	1.	.
.	.	.	.	.	.	.	.	.	.	.	1.	.	1.	.	.	.	.	.	.	-4.	1.	1.	-1.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1.	-4.	1.	1.	.	.	.	.	.	.
											-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.	.	.
											.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.	.	.
											.	.	-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.	.	.
											.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.	.	.
											.	.	.	.	-1.	.	1.	.	1.	.	.	.	.	1.	1.	-4.	1.	1.	-1.
											.	.	.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.	1.	1.
											.	.	.	.	.	.	-1.	.	1.	.	.	.	.	.	.	1.	1.	-4.	1.
	.	.	.	.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.	.	.	.	.	.	-1.	1.	1.	-4.

	1	3	5	7	9
	2	4	6	8	10
	11	13	15	17	19
	12	14	16	18	20
	21	23	25	27	29
	22	24	26	28	30

Fig.I-12

Matrix B for a Simply Supported Plate With Odd Number of Divisions Along Its Width.

TIME, 1 MINUTE AND 36 SECONDS

22.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.									
-8.	21.	2.	-8.	.	1.	.	.	.	.	.	.	-8.	1.	2.	.	.	.	.	.	.									
-8.	2.	21.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.									
2.	-8.	-8.	20.	2.	8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.									
1.	.	-8.	2.	21.	8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.									
.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.	.	.									
.	.	1.	.	-8.	2.	21.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.									
.	.	.	1.	2.	8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	.	-8.	1.	2.									
.	.	.	.	1.	.	-8.	2.	21.	-8.	-8.	2.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	.	.	.	2.	.	-8.	1.	2.	.	.	.	.	.	.	.
.	.	.	.	.	.	1.	.	-8.	2.	22.	-8.	.	.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	1.	2.	-8.	-8	21.	.	.	.	.	.	.	.	.	2.	.	-8.	1.	.	.	.	.	.	.
1.	-8.	.	2.	.	.	.	.	.	.	.	.	21.	-8.	-8.	2.	1.	.	.	.	.	.	.	.	1.	.	.	.	.	.
.	1.	.	.	.	.	.	.	.	.	.	.	-8.	21.	2.	-8.	.	1.	.	.	.	.	.	.	-8.	2.	.	.	.	.
.	2.	1.	-8.	.	2.	.	.	.	.	.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	.	1.	.	.	.	.
.	.	.	1.	.	.	.	.	.	.	.	.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	.	2.	-8.	2.	.	.	.
.	.	.	2.	1.	8.	.	2.	.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.	.
.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	.	2.	-8.	2.	.	.
.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	1.	.	.	.	.	.	1.	.	.
.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	1.	.	.	2.	-8.	2.	.
.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	.	-8.	2.	20.	-8.	-8.	2.	.	.	.	.	1.	.
.	.	.	.	.	.	.	.	.	1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	20.	2.	-8.	.	.	.	2.	-8.	2.
.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	.	.	.	1.	.	-8.	2.	21.	-8.	.	.	.	.	.	1.
.	.	.	.	.							1.	.	.	.	.	.	.	.	1.	2.	-8.	-8.	21.	.	.	.	.	2.	-8.
											.	1.	-8.	.	2.	.	.	.	.	.	.	.	.	22.	-8.	1.	.	.	.
										.	.	.	2.	1.	-8.	.	2.	.	.	.	.	.	.	-8.	21.	-8.	1.	.	.
										.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	.	1.	-8.	21.	-8.	1.	.
										.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	.	1.	-8.	21.	-8.	1.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	2.	.	.	1.	-8.	21.	-8.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2.	1.	-8.	.	.	.	1.	-8.	22.

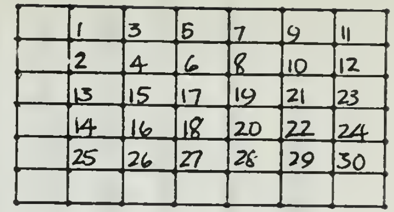


Fig. I-13

A^2-Matrix of a Clamped Plate With Even Number of Divisions in the Y-Direction.

APPENDIX II

DESCRIPTION OF THE PROGRAM

II-A. The Main Program.

The main program follows closely the developments in Appendix I. Because of the regular pattern of the matrices resulting from the scheme of numbering the mesh nodes, the program was developed for variable orders of matrices whose forms depend on the number of interior mesh nodes used and the number of divisions in the y-direction (whether they are odd or even). The position of the elements along lines parallel to the main diagonal followed simple arithmetic regularity so that the generation of the elements was in terms of the diagonals they belonged to.

Matrix A was shown to be symmetrical. This made it possible to generate the elements diagonally opposite across the main diagonal at the same time. This is not true in the case of matrix B because of the method in approximating it.

The program follows the steps enumerated below. It is summarized in a general flow chart of the program on page 52.

1. Generate A
2. Square A
3. Correct A^2 if the plate is clamped
4. Correct A^2 if lines of symmetry are used
5. Invert A^2
6. Generate matrix B
7. Correct B if lines of symmetry are used
8. Calculate $C = A^{-2}B$

9. Find the eigenvalue of C

10. Compute the buckling coefficient K

11. Repeat all of the above to get enough K's for extrapolation

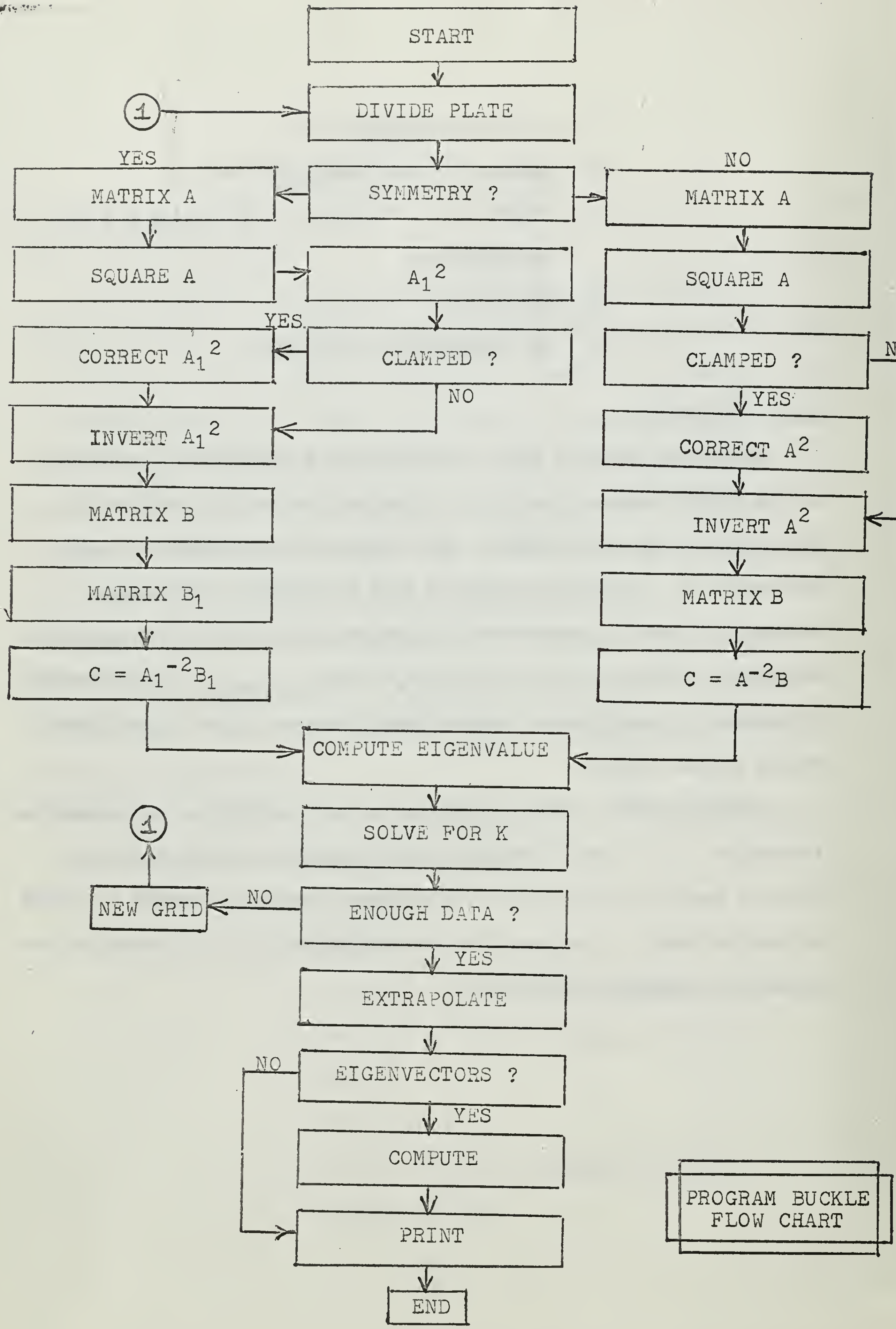
12. Extrapolate

13. Get eigenvectors if desired

II-B. Subroutines

Subroutine MATALG - This is available as a mathematical subroutine at the USNPGS Computer Facility. It has two options that suit the requirements of the main program. The first option is needed to invert the matrix A^2. The second option is used to solve the simultaneous equations to get the eigenvectors if desired, and to solve the correction equations to obtain $K_1, K_2, K_3, K_4, K_5, K_6$, and $K_{corrected}$. This subroutine is capable of providing an inverted matrix accurate to at least 9 significant decimal digits.

Subroutine EIG3- This subroutine is also available as a mathematical subroutine. It is used to evaluate the eigenvalues for the matrix C. Since it does not solve for the eigenvectors, subroutine MATALG is called to provide them. The mathematical methods applied in this subroutine are discussed extensively in Ref. 6 .


START
DIVIDE PLATE
1
SYMMETRY ?
YES
NO
MATRIX A
SQUARE A
$A_1{}^2$
CLAMPED ?
YES
NO
CORRECT $A_1{}^2$
INVERT $A_1{}^2$
MATRIX B
MATRIX B_1
$C = A_1{}^{-2}B_1$
MATRIX A
SQUARE A
CLAMPED ?
YES
NO
CORRECT A^2
INVERT A^2
MATRIX B
$C = A^{-2}B$
COMPUTE EIGENVALUE
SOLVE FOR K
ENOUGH DATA ?
NO
NEW GRID
1
YES
EXTRAPOLATE
EIGENVECTORS ?
NO
YES
COMPUTE
PRINT
END
PROGRAM BUCKLE FLOW CHART

APPENDIX III

INSTRUCTIONS ON THE USAGE OF THE PROGRAM

General.

The program is written in Fortran 60 (F-60). However, it is used as an F-63 program for three reasons. First, extra storage is obtained with the use of control cards "RELOCOM" and "EXECUTER". Second, the variable dimensioning feature is necessary to get enough computed values of the buckling coefficient for different mesh sizes for the purpose of extrapolation. Third, the time for compilation and execution is reduced with the use of binary decks. It may be noted that one minute is saved when a listing for the program is not called for.

III-A. Purpose.

Program Buckle was written to compute the initial buckling load of rectangular plates and, if desired, to find the relative deflections of point on the plate at the start of buckling. This program is limited to plates with edges that are simply supported or to plates with clamped edges.

III-B. Input Requirements.

It is assumed that the stresses at every mesh node are known and stored as one dimensional arrays in XFORCE, FY, and FXY (these correspond to F_x, F_y, and F_{xy}, respectively). In the sample program, the computation of the stresses was incorporated as part of the main program. This part can be removed completely and introduced as a subprogram. For one particular manner of dividing the plate one data card is required.

The data card is divided into 7 fields of four characters each. These are reserved for the following parameters:

1. Q = symmetry parameter. For the whole plate it is entered as 2.0. For a quarter of the plate it is entered as 1.0.

2. Clamp = support parameter. For a simply supported plate it is entered as 1.0. For clamped plates it is entered as 2.0.

3. MCC = the number of divisions in the y-direction. It must be right-justified and in fixed point. This may not be less than 6.

4. NNR = the number of divisions in the x-direction. It must be right-justified and in fixed point. This may not be less than 6.

5. AS = the length of the plate. It must be in floating point.

6. BS = the width of the plate. It must be in floating point.

7. VECTOR = eigenvector option parameter. This is entered as 1.0 when eigenvectors are needed, otherwise any other positive floating point number is entered.

Since six computed values of K are needed to extrapolate for the buckling coefficient, six data cards must be prepared. Because the program is limited to handle a maximum of 99 internal mesh nodes when the whole plate is used, the product (MCC - 1) (NNR - 1) may not exceed 99.

III-C. Output of the Program.

The program will have the following output.

1. The trace of the matrix C.

2. The iterates in finding the eigenvalue and the number of iterations.

3. The first, second, and third derivatives of the polynomial used to approximate the determinant for a given eigenvalue.

All of the above are provided by subroutine EIG 3.

4. The extrapolated value of the buckling coefficient K_1.

5. The constants used to evaluate the error, K_2, K_3, K_4, K_5 and K_6.

6. The number of divisions in the x and y directions.

7. The computed value of the buckling coefficient K for each choice of grid.

8. The corrected values in (7), $K_{corrected}$.

The values in (4) and (8) should agree very closely if the extrapolation is correct.

9. The eigenvectors are printed out when the eigenvector option is selected. The components are printed out starting with w_2 since w_1 = 1 in the program. The printout reads from left to right at the start of every line.

III-D. Cautions to Users.

The use of a quarter of a plate when the buckling is symmetrical has not been tested satisfactorily. While the matrices generated using this feature of the program were found correct, finding the eigenvalues for the matrix C required excessive computer times. Unless further tests are made, it is suggested that using the whole plate is a more reliable feature of the program to use.

When solving problems with this program, it will be important to check the iterates and the number of iterations to find the eigenvalue for every C-matrix. The subroutine for finding the eigenvalue was written to accept as an eigenvalue the 16th iterate when convergence is slow. This means that an eigenvalue which requires 16 iterations to find is not

the true eigenvalue although it may be close to it. The computed buckling coefficient based on this should be considered unreliable.

There is a theoretical basis which will not be discussed here for one to expect that the C-matrix for cases involving pure shear has a trace equal to zero. Hence, when one is dealing with such a problem he should be wary of data that is based on a C-matrix with a trace that is nonzero. For this purpose a trace whose absolute value is equal to or less than 1×10^{-11} is considered zero.

When the above difficulties are encountered, one may decide to change grids and replace the unreliable data. It may be more expedient to use the extrapolation formula with the fourth order terms omitted if enough data is left after discarding the unreliable results. In this event, at least three results must be available to use the extrapolation formula including only the second order terms.

IV-E. Time Requirements.

The program requires an average time of 3 minutes to compile and 12 minutes to execute for a total of 15 minutes running time to solve a problem using six different grids.

APPENDIX IV

SAMPLE PROGRAM

PROGRAM BUCKLE, DECK ASSEMBLY

```
CONTROL CARDS
PROGRAM BUCKLE



END
SUBROUTINE MATALG



END
SUBROUTINE EIG3



END
    END
    FINIS
    EXECUTER.
DATA CARDS
```

LIST OF SYMBOLS

A,B	arrays containing the elements of the coefficient matrices
AS, BS	length and width of the plate, respectively
AA	H_y/H_x
BB	H_x/H_y
CLAMP	edge condition parameter
FNCR	calculated buckling coefficient
FORCE1	corrected buckling coefficient
FXY	array for shear stress
FY	array for stress in the y-direction
K1	number of pairs of horizontal mesh lines. (A single last line counts as a "pair".)
NM	number of interior mesh nodes, also order of the matrices
Q	symmetry condition parameter
VECTOR	option parameter for eigenvectors
XFORCE	array for stress in the x-direction

```
-COOP,, DUMLAO MS BOX D,S/1S/2S/E/6=51,15,90000,4.
-BINARY,56.
(RELOCOM.
-FTN,L,E.
      PROGRAM MATRIX
      DIMENSION A(99,99),B(99,99),RTR(1),RTI(1),Y(99),XFORCE(99),FXY(99)
     1,FY(99),X(6 ),W(6 ),FORCE(6 ),FORCE1(6,6),C(6,6)   ,HHY(6),HHX(6)
      COMMON A,RTR,RTI,B,XFORCE,Y,FY,FXY,C
      RUN=1.
  814 READ 815, Q,CLAMP,MCC,NNR,AS,BS,VECTOR
  815 FORMAT(F4.0,F4.0,I4,I4,F4.0,F4.0,F4.0)
      NRUN=RUN
      X(NRUN)=NNR
      W(NRUN)=MCC
 7999 IF(Q-1.)222,333,222
  333 NR=NNR
      MC=MCC
      FMCC=MCC
      FNNR=NNR
      HX=AS/(2.*(FNNR-1.))
      HY=BS/(2.*(FMCC-1.))
      GO TO 99813
  222 NR=NNR-1
      MC=MCC-1
  813 HX=AS/FLOATF(NNR)
      HY=BS/FLOATF(MCC)
99813 MMC=MC/2
      NM=NR*MC
      IF(MC-1) 1,2,1
    1 NT=2*NR
      GO TO 3
    2 NT=NR
    3 IF(MMC*2-MC)4,5,4
    4 K1=(MC+1)/2
      GO TO 6
    5 K1=MC/2
    6 IF(K1-1) 7,8,7
    7 K2=(K1-1)*NT
      GO TO 9
    8 K2=K1*NT
    9 NN=NM-1
      NT3=NT-3
      NR1=NR-1
      NT4=NT+3
      K3=NT+1
      K4=NT-1
      K5=K2+1
      K6=K1-1
      K7=K2-1
      K8=NM-2
      K9= (K1-2)*NT
      K10=K9-1
      K11=K2+3
      K13=K2-3
      K22=K2+2
      NNT=NT-2
```

```
      AA=HY/HX
      BB=HX/HY
      CC=-2.*(AA+BB)
      ALPHA=2.
      MC1=MC-1
      MC2=MC-2
      DO 816 J=1,NM
      XFORCE(J)=0.0
      FXY(J)=0.0
      FY(J)=0.0
      DO 816 I=1,NM
      A(I,J)=0.0
      B(I,J)=0.0
      Y(I)=0.0
  816 CONTINUE
      RTR(1)=0.0
      DO110 J=1,NM
  110 B(J,J)=CC
      IF(MMC*2-MC)12,11,12
   11 DO 13 J=2,NM,2
   13 B(J-1,J)=BB
      GO TO 116
   12 DO 14 J=2,K2,2
      B(J,J-1)=BB
   14 B(J-1,J)=BB
      IF(Q-1.)150,114,150
  150 DO 15 J= K2,K8
      B(J+1,J+2)=AA
   15 B(J+2,J+1)=AA
      GO TO 140
  114 DO 300 J=K11,NM
  300 B(J+1,J)=AA
      DO 5151 J=K2,K8
      B(J+1,J+2)=AA
 5151 B(J+2,J+1)=AA
      B(K2+1,K2+2)=2.*AA
  140 IF(MMC*2-MC)118,116,118
  116 DO 170 L=1,K1
      IF (Q-1.)171,17,171
  171 J=(L-1)*NT+3
      B(J-2,J)=AA
      GO TO 170
   17 J=(L-1)*NT+3
      B(J-2,J)=2.*AA
  170 CONTINUE
      DO181 L=1,K1
      J=(L-1)*NT+4
      IF(Q-1.)180,18,180
  180 B(J-2,J)=AA
      GO TO 181
   18 B(J-2,J)=2.*AA
  181 CONTINUE
      GO TO 121
  118 DO 172 L= 1, K6
      J= (L-1)*NT+3
      IF (Q-1.) 177,117,177
```

```
177 B(J-2,J)=AA
    GO TO 172
117 B(J-2,J)=2.*AA
172 CONTINUE
    DO 252 L= 1,K6
    J=(L-1)*NT+4
    IF(Q-1.)250,25,250
250 B(J-2,J)=AA
    GO TO 252
 25 B(J-2,J)=2.*AA
252 CONTINUE
    GO TO 122
121 DO119 L=1,K1
    DO 20 K=5,NT
    J=(L-1)*NT+K
 20 B(J-2,J)=AA
    DO 19 K=3,NT
    J=(L-1)*NT+K
 19 B(J,J-2)=AA
119 CONTINUE
    GO TO 221
122 DO 123 L= 1,K6
    DO 124 K= 5,NT
    J=(L-1)*NT+K
124 B(J-2,J)=AA
    DO 125 K=3,NT
    J=(L-1)*NT+K
125 B(J,J-2)=AA
123 CONTINUE
    GO TO 22
221 DO 23 J=1,K7,2
    B(J+1,NT+J)=BB
 23 B(NT+J,J+1)=BB
    GO TO 26
 22 DO 224 J=1,K10,2
    B(J+1,NT+J)=BB
224 B(J+NT,J+1)=BB
    DO 225 J= 1,NR
225 B(K9+2*J ,K2+J)= BB
    IF (Q-1.)251,227,251
251 DO 253 J= 1,NR
253 B(K2+J,K9+2*J)=BB
    GO TO 800
227 DO 226 J=1,NR
226 B(K2+J,K9+2*J)=2.*BB
800 GO TO 900
 26 DO 27 J= 1,K7,2
 27 B(J+1,J)=BB
    DO282 J= K5,NN,2
    IF(Q-1.)280,28,280
280 B(J+1,J)=BB
    GO TO 282
 28 B(J+1,J  )=2.*BB
282 CONTINUE
900 DO 400 I=1,NM
    DO 305 K=1,NM
```

```
      SUM=0.
      DO 10 J=1,NM
      SUMS=SUM+B(I,J)*B(J,K)
   10 SUM=SUMS
  305 A(I,K)=SUM
  400 CONTINUE
      IF(CLAMP)611,601,611
C     THE FOLOWING IS A MODIFICATION OF THE MATRIX PREVIOUSLY
C     GENERATED WHEN THE CASE TREATED  IS ONE OF CLAMPED EDGES
  601 DO 602 I=2,NR1
      J=2*I-1
  602 A(J,J)=A(J,J)+2.*(BB)**2
      IF(Q-1.) 622,611,622
  622 IF(MMC*2-MC)662,603,662
  662 DO 610 J= K22,NN
      A(J,J)=A(J,J)+2.*(BB)**2
  610 CONTINUE
      GO TO 604
  603 DO 633 I=2,NR1
      A(K2+2*I,K2+2*I)=A(K2+2*I,K2+2*I)+2.*(BB)**2
  633 CONTINUE
  604 DO 605 L=2,K1
      J=(L-1)*NT
      A(J,J)=A(J,J)+2.*(AA)**2
  605 CONTINUE
      IF(MMC*2-MC)667,666,667
  667 DO 668 L=2,K6
      J=(L-1)*NT+K4
      A(J,J)=A(J,J)+2.*(AA)**2
  668 CONTINUE
      DO 669 L=2,K6
      J=(L-1)*NT+1
      A(J,J)=A(J,J)+2.*(AA)**2
  669 CONTINUE
      GO TO 670
  666 DO 606 L=2,K1
      J=(L-1)*NT+K4
      A(J,J)=A(J,J)+2.*(AA)**2
  606 CONTINUE
      DO 607 L=2,K1
      J=(L-1)*NT+1
      A(J,J)=A(J,J)+2.*(AA)**2
  607 CONTINUE
  670 DO 618 L=1,K6
      J=(L-1)*NT+2
      A(J,J)=A(J,J)+2.*(AA)**2
  618 CONTINUE
      A(1,1)=A(1,1)+2.*((AA)**2+(BB)**2)
      A(K4,K4)=A(K4,K4)+2.*((AA)**2+(BB)**2)
      A(NM,NM)=A(NM,NM)+2.*((AA)**2+(BB)**2)
      IF(MMC*2-MC)608,609,608
  609 A(K22,K22)=A(K22,K22)+2.*((AA)**2+(BB)**2)
      GO TO 611
  608 A(K5,K5)=A(K5,K5)     +2.*((AA)**2+(BB)**2)
C     THE FOLLOWING CORRECTS THE ABOVE WHEN LINES OF SYMMETRY ARE USED
  611 IF(Q-1.)3000,3001,3000
```

```
3001 IF(MMC*2-MC)3002,3006,3002
3002 DO 3003 J=1,NM
     DO 3003 L=1,K6
     I=(L-1)*NT+1
     A(I,J)=.5*A(I,J)
3003 CONTINUE
     DO 3004 J=1,NM
     DO 3004 L=1,K6
     I=(L-1)*NT+2
     A(I,J)=.5*A(I,J)
3004 CONTINUE
     DO 3055 J=1,NM
     DO 3005 K=2,NR
     A(K2+K,J)=.5*A(K2+K,J)
3005 CONTINUE
     A(K5,J)=.25*A(K5,J)
3055 CONTINUE
     GO TO 3000
3006 DO 3007 J=1,NM
     DO 3007 L=1,K1
     I=(L-1)*NT+1
     A(I,J)=.5*A(I,J)
3007 CONTINUE
     DO 3008 J=1,NM
     DO 3008 L=1,K6
     I=(L-1)*NT+2
     A(I,J)=.5*A(I,J)
3008 CONTINUE
     DO 3009 J=1,NM
     DO 3099 I=2,NR
     A(K2+2*I,J)=.5*A(K2+2*I,J)
3099 CONTINUE
     A(K5+1,J)=.25*A(K5+1,J)
3009 CONTINUE
3000 DO 1000 I=1,NM
     DO 1000 J=1,NM
     B(I,J)=0.0
1000 CONTINUE
C    FINDING THE INVERSE OF MATRIX A-SQURED
     CALL MATALG (A,B,NM,NM,1,DET,99)
     DO 90 J=1,NM
     DO 90 I= 1,NM
  90 A(I,J)=0.
     IF(MMC*2-MC)91,94,91
  91 DO92 N=1,MC2,2
     DO92 I=1,NR
     J=(N-1)*NR+2*I-1
     FA=N
     FB=MCC
     FL=FA/FB
     FNA=NNR
     FIA=I
     FIX=FIA/FNA
     FXY(J)=0.0
     FY(J)=0.0
     XFORCE(J)=1.0
```

```
 92 CONTINUE
    DO93 N=2,MC1,2
    DO93 I=1,NR
    J=(N-2)*NR+2*I
    FA=N
    FB=MCC
    FL=FA/FB
    FIA=I
    FNA=NNR
    FIX=FIA/FNA
    FXY(J)=0.0
    FY(J)=0.0
    XFORCE(J)=1.0
 93 CONTINUE
    DO 933 I=1,NR
    J=K2+I
    FIA=I
    FNA=NNR
    FIX=FIA/FNA
    FMC=FLOATF(MC)
    FMCC=FLOATF(MCC)
    FMMCC=FMC/FMCC
    FXY(J)=0.0
    FY(J)=0.0
    XFORCE(J)=1.0
933 CONTINUE
    DO953 J= 1, NM
953 A(J,J)=-2.*(AA*XFORCE(J)+BB*FY(J))
 80 DO 82 L=1,K6
    DO 82 I=3,NT
    J=(L-1)*NT+I
    A(J-2,J)=AA*XFORCE(J-2)
 82 A(J,J-2)=AA*XFORCE(J)
    K12=K2+2
    DO 83 J=K12,NM
    A(J-1,J)=AA*XFORCE(J-1)
 83 A(J,J-1)=AA*XFORCE(J)
    GO TO 5551
 94 DO95 N=1,MC1,2
    DO95 I=1,NR
    J=(N-1)*NR+2*I-1
    FA=N
    FB=MCC
    FL=FA/FB
    FIA=I
    FNA=NNR
    FIX=FIA/FNA
    FY(J)=0.0
    XFORCE(J)=1.0
    FXY(J)=0.0
 95 CONTINUE
    DO96 N=2,MC,2
    DO96 I=1,NR
    J=(N-2)*NR+2*I
    FA=N
    FB=MCC
```

001740

```
      FL=FA/FB
      FIA=I
      FNA=NNR
      FIX=FIA/FNA
      FXY(J)=0.0
      FY(J)=0.0
      XFORCE(J)=1.0
   96 CONTINUE
      DO954 J= 1, NM
  954 A(J,J)=-2.*(AA*XFORCE(J)+BB*FY(J))
   81 DO 86 L=1,K1
      DO 86 K= 3,NT
      J=(L-1)*NT+K
      A(J,J-2)=AA*XFORCE(J)
   86 A(J-2,J)=AA*XFORCE(J-2)
 5551 IF(Q-1.)3010,3011,3010
 3011 IF(MMC*2-MC)3012,3013,3012
 3012 DO 3014 L=1,K6
      I=(L-1)*NT+3
      A(I-2,I)=2.*XFORCE(I-2)
 3014 CONTINUE
      DO 3015 L=1,K6
      I=(L-1)*NT+4
      A(I-2,I)=2.*XFORCE(I-2)
 3015 CONTINUE
      A(K5,K5+1)=2.*XFORCE(K5)
      GO TO 3010
 3013 DO 3016 L=1,K1
      I=(L-1)*NT+3
      J=(L-1)*NT+4
      A(I-2,I)=2.*XFORCE(I-2)
      A(J-2,J)=2.*XFORCE(J-2)
      A(K5+1,K5+3)=2.*XFORCE(K5+1)
 3016 CONTINUE
C      GENERATION OF MATRIX DUE TO SHEAR
 3010 IF(K2-NT)5000,5001,5000
 5001 IF(MCC-2)5002,5003,5002
C      GENERATION FO THE MATRIX DUE TO SHEAR WHEN MCC IS ODD
 5003 DO918 J=4,NT,2
      A(J-3,J)=-FXY(J-3)/2.
  918 A(J,J-3)=-FXY(J)/2.
      DO919 J=3,K4,2
      A(J-1,J)=FXY(J-1)/2.
  919 A(J,J-1)=FXY(J)/2.
      GO TO 5004
 5002 IF(MC-3)5000,5005,5000
 5005 DO 5006 J=1,NR1
      A(2*J,K7+J)=-FXY(2*J)/2.
      A(K7+J,2*J)=-FXY(K7+J)/2.
      A(2+2*J,K2+J)=FXY(2+2*J)/2.
 5006 A(K2+J,2+2*J)=FXY(K2+J)/2.
      GO TO 5007
 5000 IF(MMC*2-MC)5008,5009,5008
 5008 DO966J=NT4,K7,2
      A(J-K3,J)=-FXY(J-K3)/2.
  966 A(J,J-K3)=-FXY(J)/2.
```

```
      DO977J=K3,K13,2
      A(J-NT3,J)= FXY(J-NT3)/2.
  977 A(J,J-NT3)= FXY(J)/2.
      DO988J=1,NR1
      I=K2-NT+2+2*J
      A(I,K2+J)= FXY(I)/2.
      A(K2+J,I)= FXY(K2+J)/2.
      L=K2-NT+2*J
      M=K2+1
      A(L,M+J)=-FXY(L)/2.
  988 A(M+J,L)=-FXY(M+J)/2.
 5007 DO3120L=1,K6
      DO1211 K=4,NT,2
      J=(L-1)*NT+K
      A(J-3,J)=-FXY(J-3)/2.
 1211 A(J,J-3)=-FXY(J)/2.
 3120 CONTINUE
      DO1140L=1,K6
      DO1130K=3,K4,2
      J=(L-1)*NT+K
      A(J-1,J)=FXY(J-1)/2.
 1130 A(J,J-1)=FXY(J)/2.
 1140 CONTINUE
      K33=K1-3
      IF(K33)34413,34413,34414
34414 DO 14413 J=1,K33
      I=J*NT
      A(I,I+K3)=0.0
      A(I+K3,I)=0.0
      A(I+2,K3-2)=0.0
      A(K3-2,I+2)=0.0
14413 CONTINUE
34413 IF(Q-1.)5004,700,5004
  700 DO 408 L=2,K6
      J=(L-1)*NT+3
  408 A(J-K3,J)=0.
      DO 409 L=1,K6
      J=(L-1)*NT+4
  409 A(J-3,J)=0.
      DO 410 L=1,K6
      J=(L-1)*NT+3
  410 A(J-1,J)=0.
      I=(K1-2)*NT+2
      J=K2+2
      A(I,J)=0.
      K14=(K1-2)*NT+4
      DO 411 J=4,K14,NT
  411 A(J+NT3,J)=0.
      DO 412 J=1,NR1
      L=K2-NT+2*J
      M=K2+1
  412 A(M+J,L)=0.
      DO 413 J=1,NR1
      I=K2-NT+2+2*J
  413 A(K2+J,I)=0.
      A(K2+1,K14)=0.
```

```
 5004 GO TO 5555
 5009 DO1660L=2,K1
      DO1660K=3,K4,2
      J=(L-1)*NT+K
      A(J-K3,J)=-FXY(J-K3)/2.
 1660 A(J,J-K3)=-FXY(J)/2.
      DO1770L=2,K1
      DO1770K=1,NT3,2
      J=(L-1)*NT+K
      A(J-NT3,J)= FXY(J-NT3)/2.
 1770 A(J,J-NT3)= FXY(J)/2.
      DO1111 L=1,K1
      DO1111 K=4,NT,2
      J=(L-1)*NT+K
      A(J-3,J)=-FXY(J-3)/2.
 1111 A(J,J-3)=-FXY(J)/2.
      DO1114 L=1,K1
      DO1113 K=3,K4,2
      J=(L-1)*NT+K
      A(J-1,J)=FXY(J-1)/2.
 1113 A(J,J-1)=FXY(J)/2.
 1114 CONTINUE
      K33=K1-3
      IF(K33)44413,44413,44414
44414 DO 24413 J=1,K33
      I=J*NT
      A(I+K3,I)=0.0
      A(I,I+K3)=0.0
      A(I+2,K3-2)=0.0
      A(K3-2,I+2)=0.0
24413 CONTINUE
44413 IF(Q-1.)5004,5010,5004
 5010 DO 401 L=2,K1
      J=(L-1)*NT+3
  401 A(J-K3,J)=0.
      DO 402 L=1,K5
      J=(L-1)*NT+4
  402 A(J-3,J)=0.
      DO 403 L=1,K6
      J=(L-1)*NT+3
  403 A(J-1,J)=0.
      DO 405 J=K5 ,NN,2
  405 A(J-1,J)=0.
      DO 406 L=1,K6
      J=(L-1)*NT+4
  406 A(J+NT3,J)=0.
      DO 440 J=1,NR1
      I=K2+2+2*J
      M=K2+2*J-1
  440 A(I,M)=0.
      A(K2+2,K2+3)=0.
C     GENERATION OF MATRIX DUE TO FY WHEN MCC IS ODD
 5555 IF (MMC*2-MC)2112,2113,2112
 2112 DO2114 J=2,K2,2
      A(J-1,J)=BB*FY(J-1)
 2114 A(J,J-1)=BB*FY(J)
```

```
      GO TO 2115
 2113 DO 2116 J=2,NM,2
      A(J-1,J)=BB*FY(J-1)
 2116 A(J,J-1)=BB*FY(J)
      DO2150J=K3,NN,2
      A(J,J-K4)=BB*FY(J)
 2150 A(J-K4,J)=BB*FY(J-K4)
      IF (Q-1.)2115,2117,2115
 2117 DO2118J= K5,NN,2
 2118 A(J+1,J)=2.*BB*FY(J+1)
 2115 IF(K2-NT)2119,2120,2119
 2120 IF(MMC*2-MC)2121,2122,2121
 2121 DO2123 J=1,NR
      A(J*2,K2+J)=BB*FY(J*2)
 2123 A(K2+J,J*2)=BB*FY(K2+J)
      GO TO 2122
 2119 IF(MMC*2-MC)2124,2125,2124
 2125 DO2126J=K3,NN,2
      A(J-K4,J)=BB*FY(J-K4)
 2126 A(J,J-K4)=BB*FY(J)
      IF(Q-1.)2127,2128,2127
 2128 DO2129 J=K5,NM
 2129 A(J,J-K4)=2.*FY(J)
      IF(Q-1.)2122,2130,2122
 2130 DO2131J=K5,K7,2
 2131 A(J+1,J)=2.*BB*FY(J+1)
 2127 GO TO 2122
 2124 DO 2132 J=K3,K7,2
      A(J-K4,J)=BB*FY(J-K4)
 2132 A(J,J-K4)=BB*FY(J)
      DO 2133 J=1,NR
      I=K2-NT+2*J
 2133 A(I,K2+J)=BB*FY(I)
 2134 DO 2135J=1,NR
      I=K2-NT+2*J
 2135 A(K2+J,I)=BB*FY(K2+J)
      IF(Q-1.)2122,999,2122
  999 DO 888 J=1,NR
      I=K2-NT+2*J
  888 A(K2+J,I)=2.*BB*FY(K2+J)
C     THE FOLLOWING IS A CORRECTION OF THE FORCE MATRIX WHEN LINES
C      OF SYMMETRY ARE USED
 2122 IF(Q-1.)4000,4001,4000
 4001 IF(MMC*2-MC)4002,4006,4002
 4002 DO 4003 J=1,NM
      DO 4003 L=1,K1
      I=(L-1)*NT+1
      A(I,J)=.5*A(I,J)
 4003 CONTINUE
      DO 4004 J=1,NM
      DO 4004 L=1,K6
      I=(L-1)*NT+2
      A(I,J)=.5*A(I,J)
 4004 CONTINUE
      DO 4055 J=1,NM
      DO 4005 K=2,NR
```

```
      A(K2+K,J)=.5*A(K2+K,J)
 4005 CONTINUE
      A(K5,J)=.25*A(K5,J)
 4055 CONTINUE
      GO TO 4000
 4006 DO 4007 J=1,NM
      DO 4007 L=1,K1
      I=(L-1)*NT+1
      A(I,J)=.5*A(I,J)
 4007 CONTINUE
      DO 4008 J=1,NM
      DO 4008 L=1,K6
      I=(L-1)*NT+2
      A(I,J)=.5*A(I,J)
 4008 CONTINUE
      DO 4009 J=1,NM
      DO 4099 I=2,NR
      A(K2+2*I,J)=.5*A(K2+2*I,J)
 4099 CONTINUE
      A(K5+1,J)=.25*A(K5+1,J)
 4009 CONTINUE
 4000 DO 901 I= 1,NM
      DO 50 K=1,NM
      SUM=0.
      DO 60 J=1,NM
      SUMS=SUM+B(I,J)*A(J,K)
   60 SUM=SUMS
   50 Y(K)=SUM
      DO320 K=1,NM
  320 B(I,K)=Y(K)
  901 CONTINUE
      DO 1001 J=1,NM
      DO 1001 I=1,NM
      A(I,J)=B(I,J)
 1001 CONTINUE
C     FINDING THE EIGENVALUE
 3999 CALL EIG3(A,NM,1,RTR,RTI,99)
C     SOLVING FOR THE BUCKLING COEEFICIENT
 1009 FNCR=FLOATF(MCC)*FLOATF(NNR)*BS/(AS*RTR(1))
      IF(Q-1.)11009,11010,11009
11009 GO TO 11011
11010 FNCR=4.*(FNNR-1.)*(FMCC-1.) *BS/(AS*RTR(1))
11011 IF(RUN-6.)1017,1016,1017
 1017 NRUN=RUN
      FORCE(NRUN)=ABSF(FNCR)
      RUN=RUN+1.
      GO TO 814
 1016 FORCE(NRUN)=ABSF(FNCR)
      PRINT 8920
 8920 FORMAT(1H1/(40X,2H .,///////////////)
      PRINT 8919,AS,BS
 8919 FORMAT(////////,40X,14H ASPECT RATIO=E10.5,1H/,E10.5)
      DO 8883 J=1,6
      FORCE1(J,1)=FORCE(J)
      C(J,1)=1.0
      HHX(J)=AS/X(J)
```

```
      HHY(J)=BS/W(J)
      C(J,2)=(HHX(J))**2
      C(J,4)=(HHY(J))**2
      C(J,3)=(HHX(J))**4
      C(J,5)=(HHY(J))**4
      C(J,6)=((HHY(J))**2)*(HHX(J))**2
 8883 CONTINUE
      CALL MATALG(C,FORCE1,6,1,0,DET,6)
      CK1=FORCE1(1,1)
      PRINT 8885,CK1
 8885 FORMAT(44X,4H K1=E20.10)
      CK2=FORCE1(2,1)
      PRINT 8886,CK2
 8886 FORMAT(44X,4H K2=E20.10)
      CK3=FORCE1(3,1)
      PRINT 8887,CK3
 8887 FORMAT(44X,4H K3=E20.10)
      CK4=FORCE1(4,1)
      PRINT 8888,CK4
 8888 FORMAT(44X,4H K4=E20.10)
      CK5=FORCE1(5,1)
      PRINT 8889,CK5
 8889 FORMAT(44X,4H K5=E20.10)
      CK6=FORCE1(6,1)
      PRINT 8890,CK6
 8890 FORMAT(44X,4H K6=E20.10)
      PRINT 8814
 88140FORMAT(//,17X,18H DIVISIONS ALONG X, 2X,18H DIVISIONS ALONG Y
     1,12X,2H K,11X,12H K,CORRECTED//)
      DO 8813 J=1,6
      FORCE1(J,1)=FORCE(J)-CK2*(AS/X(J))**2-CK3*(AS/X(J))**4-CK4*(BS/W(J
     1))**2-CK5*(BS/W(J))**4-CK6*((AS/X(J))**2)*(BS/W(J))**2
      Y1=W(J)
      X1=X(J)
      FORCE2=FORCE1(J,1)
      FORCE3=FORCE(J)
      PRINT 8915,X1,Y1,FORCE3,FORCE2
 8915 FORMAT(    15X,4E20.12)
 8813 CONTINUE
      IF(VECTOR-1.)8950,8951,8950
 8951 DO 1004 I=2,NM
      A(J-1,I-1)=B(J,I)
 1004 CONTINUE
      DO 1002 J=1,NN
 1002 A(J,J)=A(J,J)-RTR(1)
      DO 1005 J=1,NN
      A(J,NM)=-B(J+1,1)
 1005 CONTINUE
      DO 8006 J=1,NM
      DO 8006 I= 1,NM
      B(I,J)=0.0
 8006 CONTINUE
      DO 8007 J=1,NM
      B(J,1)=A(J,NM)
 8007 CONTINUE
      CALL MATALG (A,B,NN,1,0,DET,99)
```

```
      PRINT 9801
 9801 FORMAT(//,44X,13H EIGENVECTORS//)
      PRINT 9800,((B(J,I),J=1,NN),I=1,1)
 9800 FORMAT(5E20.12)
 8950 END
      SUBROUTINE MATALG(A,X,NR,NV,IDO,DET,NACT)
      DIMENSION A(NACT,NACT),X(NACT,NACT)
      IF(IDO) 1,2,1
    1 DO 3 I=1,NR
      DO 4 J=1,NR
    4 X(I,J)=0.0
    3 X(I,I)=1.0
      NV=NR
    2 DET=1.0
      NR1=NR-1
      DO 5 K=1,NR1
      IR1=K+1
      PIVOT=0.0
      DO 6 I=K,NR
      Z=ABSF(A(I,K))
      IF(Z-PIVOT) 6,6,7
    7 PIVOT=Z
      IPR=I
    6 CONTINUE
      IF(PIVOT) 8,9,8
    9 DET=0.0
      RETURN
    8 IF(IPR-K) 10,11,10
   10 DO 12 J=K,NR
      Z=A(IPR,J)
      A(IPR,J)=A(K,J)
   12 A(K,J)=Z
      DO 13 J=1,NV
      Z=X(IPR,J)
      X(IPR,J)=X(K,J)
   13 X(K,J)=Z
      DET=-DET
   11 DET=DET*A(K,K)
      PIVOT=1.0/A(K,K)
      DO 14 J=IR1,NR
      A(K,J)=A(K,J)*PIVOT
      DO 14 I=IR1,NR
   14 A(I,J)=A(I,J)-A(I,K)*A(K,J)
      DO 5 J=1,NV
      IF(X(K,J)) 15,5,15
   15 X(K,J)=X(K,J)*PIVOT
      DO 16 I=IR1,NR
   16 X(I,J)=X(I,J)-A(I,K)*X(K,J)
    5 CONTINUE
      IF(A(NR,NR)) 17,9,17
   17 DET=DET*A(NR,NR)
      PIVOT=1.0/A(NR,NR)
      DO 18 J=1,NV
      X(NR,J)=X(NR,J)*PIVOT
      DO 18 K=1,NR1
      I=NR-K
```

```
      SUM=0.0                                                           000510
      DO 19 L=I,NR1                                                     000520
   19 SUM=SUM+A(I,L+1)*X(L+1,J)                                         000530
   18 X(I,J)=X(I,J)-SUM                                                 000540
      RETURN
      END                                                               000550
      SUBROUTINE EIG3(A,N,M,RTR,RTI,NQ)                               00000000
C           474 10152 EIGENVALUES OF REAL MATRICES                    00000010
C051/1 EIGENVALUES OF NON-SYMMETRIC MATRICES                           00000020
      DIMENSIONA(NQ,NQ),NC(100),RTR(M),RTI(M)                         00000030
      CALL OVFSET                                                     0000  40
      TRACE=A(1,1)                                                    00000 50
      DO 10 I=2,N                                                     00000 60
   10 TRACE=TRACE+A(I,I)                                              00000 70
      WRITE OUTPUT TAPE 6,4,TRACE                                     00000080
      CALL TRING(A,1.E-7,N,NC, NQ)                                    00000090
      TRACE=A(1,1)                                                    00000100
      DO 11 I=2,N                                                     00000110
   11 TRACE=TRACE+A(I,I)                                              00000120
      WRITE OUTPUT TAPE 6,4,TRACE                                     00000130
      NU=0                                                            000  140
      NV=0                                                            000  150
   13 IF (NV-N)14,12,14                                               00000160
   14 NV=NV+1                                                         00000170
      NU=NV                                                           0000 180
   16 IF (NC(NV))15,17,15                                             00000190
   15 NV=NV+1                                                         0000 200
      GO TO 16                                                        00000210
   17 IF (NV-NU)19,18,19                                              00000220
 18   RTR(NU)=A(NU,NU)                                                00000230
      RTI(NU)=0.                                                      00000240
      WRITE OUTPUT TAPE 6,5,RTR(NU),RTI(NU)                           00000250
      GO TO 13                                                        00000260
   19 IF (NV-NU-1)20,21,20                                            00000270
 20   NP=XMINOF(M,NV)                                                 00000280
      CALL LAGER(A,1.E-4,NP,NU,NV,RTR,RTI,NQ)                         00000290
      GO TO 13                                                        00000300
   21 RR=.5*(A(NU,NU)+A(NV,NV))                                       00000310
      E1=RR**2-A(NU,NU)*A(NV,NV)+A(NU,NV)*A(NV,NU)                    00000320
      S=SQRTF(ABSF(E1))                                               00000330
      IF(E1)22,23,23                                                  00000340
 23   RTR(NU)=RR+S                                                    00000350
      RTI(NU)=0.                                                      00000360
      RTR(NV)=RR-S                                                    00000370
      RTI(NV)=0.                                                      00000380
 25   WRITE OUTPUT TAPE 6,5,RTR(NU),RTI(NU),RTR(NV),RTI(NV)           00000390
      GO TO 13                                                        0000 400
 22   RTR(NU)=RR                                                      00000410
      RTI(NU)=S                                                       00000420
      RTR(NV)=RR                                                      00000430
      RTI(NV)=-S                                                      00000440
      GO TO 25                                                        00000450
 12   X=0.                                                            0000 460
C     CALL FPOLD                                                      00000470
      DO 24 J=1,M                                                     00000480
 24   X=X+RTR(J)                                                      00000490
```

```
      WRITE OUTPUT TAPE 6,6,X
      RETURN
    4 FORMAT(1H048X,7HTRACE =E16.8)
    5 FORMAT (11HOEIGENVALUE 12X,2E20.8)
 6    FORMAT(1H035X,20HSUM OF EIGENVALUES =E16.8)
      END
C051/2ALMOST TRIANGULAR (HESSENBERG) SUBROUTINE
      SUBROUTINE TRING(A,EPS,N,INT, NQ)
      DIMENSION A(NQ,NQ),   INT(NQ)
      WRITE OUTPUT TAPE 6,1
      N1=N-1
      N2=N-2
      DO 21 J=1,N1
      S=ABSF(A(J,J+1))
      J1=J+1
      J2=J+2
      L=J1
      NJ1=N-J1
      IF (NJ1) 15,15,6
 6    DO 12 K=J2,N
      T=ABSF(A(J,K))
      IF(T-S)12,12,11
 11   L=K
      S=T
 12   CONTINUE
      IF(L-J1)13,15,13
13    DO 131 K=1,N
      T=A(K,J+1)
      A(K,J+1)=A(K,L)
131   A(K,L)=T
14    DO 141 K=1,N
      T=A(J+1,K)
      A(J+1,K)=A(L,K)
141   A(L,K)=T
 15   IF(S-EPS*MIN1F(ABSF(A(J,J)),ABSF(A(J+1,J+1))))16,16,17
 16   L=0
      NJ1=0
      GO TO 181
 17   T=A(J,J+1)
      DO 18 K=J2,N
 18   A(J,K)=A(J,K)/T
 181  DO 20 I=1,N
      M=MINOF(J,I-2)
      U=0.
      IF (NJ1) 19,19,7
7     DO 8 K=J2,N
 8    U=U+A(K,I)*A(J,K)
19    IF (M) 20,20,9
 9    DO 10 K=1,M
 10   U=U-A(K,I)*A(J+1,K+1)
 20   A(J+1,I)=A(J+1,I)+U
 21   INT(J)=L
 22   INT(N)=0
      RETURN
 1    FORMAT(1H048X,22HALMOST TRIANGULAR FORM)
      END
```

```
C051/3LAGUERRE METHOD
      SUBROUTINE LAGER(A,EPS,N1,NU,N,RTR,RTI,NDI)
      DIMENSION A(NDI,NDI),P(6,101),RTR(NDI),RTI(NDI),B(6)
      WRITE OUTPUT TAPE 6,1
      ONCE=0.
      BL1=1.
      NUQ=NU-1
      LLY=0
      DELOLD=1.
      ROLD=1.
      EGSUM1=0.
      EGSUM2=0.
C     CALL FPTEST(Z)
      DO 10 L=2,6
 10   P(L,NU)=0.
      NU1=NU+1
      CUP=0.
      DO 11 J=NU1,N
 11   CUP=CUP+ABSF(A(J-1,J))
      CUP=CUP/FLOATF(N-NU)
      CAP=0.
      P(1,NU)=1.
C
C     FIND TRACE OF H AND H SQUARED
C
      SPUR1=A(NU,NU)
      SPUR2=A(NU,NU)**2
      DO 13 J=NU1,N
      SPUR1=SPUR1+A(J,J)
   13 SPUR2=SPUR2+A(J,J)**2+2.*A(J-1,J)*A(J,J-1)
C
C     INITIAL ITERATE FROM INFINITY
C
   14 S1R=EGSUM1-SPUR1
      S2R=SPUR2-EGSUM2
      F1=N-NUQ
      IF(ABSF(S1R)+ABSF(S2R)-1.E-7*CAP)15,15,16
   15 XBAR=CUP
      YBAR=0.
      GO TO 23
 16   F2=F1-1.
      DR=F2*(F1*S2R-S1R**2)
      ER=SQRTF(ABSF(DR))
      IF(DR)17,18,18
   17 XBAR=-2.*S1R/F1
      YBAR=2.*ER/F1
      GO TO 23
   18 YBAR=0.
      F2=SIGNF(1.,S1R)
      IF(S1R)22,20,22
   20 F2=0.
   22 XBAR=-(S1R+F2*ER)/F1
C
C     EVALUATE POLYNOMIAL AND DERIVATIVES
C
   23 IF(ABSF(YBAR)-ABSF(XBAR)*1.E-6)24,25,25
```

```
   24 YBAR=0.
   25 M=6
      IF(YBAR)27,26,27
   26 M=3
 27   DO 34 K=NU,N
      T=-A(K,K+1)
      DO 34 L=1,M
      S=SIGNF(1.,3.5-FLOATF(L))
      M1=L+3*XFIXF(S)
      R=-XBAR*P(L,K)+YBAR*S*P(M1,K)-FLOATF(XMODF(L-1,3))*P(L-1,K)
      DO 28 J=NU,K
 28   R=R+P(L,J)*A(K,J)
      CALL OVFTST (Z)
      IF(Z)29,32,29
 29   Z=0.
      P(1,NU)=1.E-10*P(1,NU)
      IF(P(1,NU))30,30,27
 30   F=FLOATF(K-NU)/FLOATF(N-NU+1)
      WRITE OUTPUT TAPE 6,2.XBAR,YBAR,P(1,NU),F,ONCE
      XBAR=XBAR*F
      YBAR=YBAR*F
      P(1,NU)=1.0
      GO TO 27
 32   IF(N-K)33,33,34
 33   T=1.0
 34   P(L,K+1)=R/T
C
C     SCALE DOWN
C
      DO 39 K=1,6
   39 B(K)=0.
      DO 35 J=1,M
   35 B(J)=P(J,N+1)
      G1=ABSF(B(1))+ABSF(B(4))
      G2=ABSF(B(2))+ABSF(B(5))
      G3=ABSF(B(3))+ABSF(B(6))
      WRITE OUTPUT TAPE 6,2,XBAR,YBAR,G1,G2,G3
 2    FORMAT(8H ITERATE20X,E15.8,5X,E15.8,8X,3E15.4)
      D=ABSF(B(1))
      DO 36 K=2,M
   36 D=MAX1F(D,ABSF(B(K)))
      CALL SCALE(D,B,M)
      IF (G1) 41,41,43
C
C     REMOVE KNOWN ROOTS
C
   43 Q1R=0.
      Q1I=0.
      Q2R=0.
      Q2I=0.
      IF(NUQ-NU)19,21,21
   21 DO 44 J=NU,NUQ
      D1=RTR(J)-XBAR
      D2=RTI(J)-YBAR
      D=D1**2+D2**2
      D1=D1/D
```

```
      D2=-D2/D
      Q1R=Q1R+D1
      Q1I=Q1I+D2
      Q2R=Q2R+D1**2-D2**2
   44 Q2I=Q2I+2.*D1*D2
C
C     FIND S1 AND S2
C
   19 T1R=B(2)/B(1)
      T1I=0.
      T2R=B(3)/B(2)
      T2I=0.
      IF(YBAR)45,46,45
   45 D1=B(1)**2+B(4)**2
      D2=B(2)**2+B(5)**2
      T1R=(B(2)*B(1)+B(5)*B(4))/D1
      T1I=(B(5)*B(1)-B(4)*B(2))/D1
      T2R=(B(3)*B(2)+B(6)*B(5))/D2
      T2I=(B(6)*B(2)-B(5)*B(3))/D2
   46 S1R=T1R+Q1R
      S1I=T1I+Q1I
      S2R=T1R*(T1R-T2R)-T1I*(T1I-T2I)-Q2R
      S2I=T1R*(T1I-T2I)+T1I*(T1R-T2R)-Q2I
C
C     FIND THE NEXT ITERATE
C
      LLY=LLY+1
      D=ABSF(XBAR)+ABSF(YBAR)
      IF(1.E+7-D*(ABSF(S1R)+ABSF(S1I))) 41,41,42
   41 MARK=1
      GO TO 100
   42 G=N-NUQ
   48 IF(YBAR-ABSF(X))50,50,49
   49 S1I=S1I+1./(2.*YBAR)
      S2R=S2R+1./(4.*YBAR**2)
      G=G-1.
 50   IF(BL1)65,65,66
 65   H=.5*(G-2.)
      GO TO 67
 66   H=G-1.
 67   DR=H*(G*S2R-S1R**2+S1I**2)
      DI=H*(G*S2I-2.*S1R*S1I)
      IF(DI)53,51,53
   51 EI=0.
      ER=SQRTF(ABSF(DR))
      IF(DR)52,54,54
   52 EI=ER
      ER=0.
      GO TO 54
   53 CALL CXSQRT(DR,DI,ER,EI)
   54 IF(S1R*ER+S1I*EI)55,56,56
   55 ER=-ER
      EI=-EI
   56 D1=S1R+ER
      D2=S1I+EI
      D=D1**2+D2**2
```

```
      X=-G*D1/D
      XBAR=XBAR+X
      Y=G*D2/D
      YBAR=YBAR+Y
      DELNEW=ABSF(X)+ABSF(Y)
      RNEW=DELNEW/DELOLD
      D=ABSF(XBAR)+ABSF(YBAR)
C
C     TEST FOR LINEAR CONVERGENCE
C
      IF(LLY-3)62,62,57
 57   IF (DELNEW-MAX1F(3.*DELOLD,.5*D))571,571,570
 570  IF (BL1) 571,571,572
 572  DELOLD=CAP
      ROLD=3.
      IF (LLY-15) 14,14,100
 571  IF(RNEW-.7*ROLD) 62,58,58
   58 MARK=3
      IF(DELNEW-.001*EPS*CAP) 70,59,59
 59   IF(BL1)61,61,60
 60   XBAR=XBAR-X
      YBAR=YBAR-Y
      BL1=0.
      GO TO 48
 61   BL1=1.
      GO TO 63
C
C     TEST FOR AN EIGENVALUE
C
   62 IF(DELNEW-EPS*MAX1F(D,.001*CAP))64,64,63
 63   DELOLD=DELNEW
      ROLD=RNEW
      IF (LLY-15) 23,23,100
C
C     DO WE HAVE A COMPLEX APPROACH TO A REAL ROOT
C
   64 MARK=2
   70 BL1=1
      IF(YBAR)71,100,71
   71 IF(G2*ABSF(YBAR)-G1)72,100,100
   72 IF(ONCE)73,73,100
   73 X=0.
      ONCE=1.
      YBAR=0.
      GO TO 63
C
C     WE ACCEPT (XBAR,YBAR) AS A ROOT
C
  100 NUQ=NUQ+1
      RTR(NUQ)=XBAR
      IF(ABSF(YBAR)-.001*ABSF(XBAR))74,74,75
   74 YBAR=0.
   75 IF(NUQ-NU) 9,76, 9
    9 IF(RTI(NUQ-1))76,76,77
   76 YBAR=ABSF(YBAR)
      RTI(NUQ)=YBAR
```

```
      GO TO 78
   77 RTI(NUQ)=-ABSF(YBAR)
   78 WRITE OUTPUT TAPE 6,3,RTR(NUQ),RTI(NUQ),LLY,MARK
    3 FORMAT(11HOEIGENVALUE12X,2E20.8,12X,I3,17H ITERATIONS,TEST I1//)
      LLY=0
      CAP=MAX1F(D,CAP)
      DELOLD=1.
      ROLD=1.
      EGSUM1=EGSUM1+RTR(NUQ)
      EGSUM2=EGSUM2+RTR(NUQ)**2-RTI(NUQ)**2
      IF(NUQ-N1)60,101,101
   80 IF(YBAR)83,84,81
   81 YBAR=-YBAR
      GO TO 23
   83 IF(NUQ-NU)31,84,31
   31 RTI(NUQ-1)=.5*(RTI(NUQ-1)-RTI(NUQ))
      RTI(NUQ)=-RTI(NUQ-1)
C
C     A NEWTON ITERATE TOWARDS NEXT ROOT
C
   84 ONCE=0.
      Z=0.
      IF((ABSF(Q1R)+ABSF(Q1I))*D-10000.)85,85,14
 85   IF(ABSF(EGSUM1-SPUR1)+ABSF(EGSUM2-SPUR2)-1.E-5*CAP)15,15,86
   86 DR=B(3)+2.*(B(2)*Q1R-B(5)*Q1I)
      DI=B(6)+2.*(B(2)*Q1I+B(5)*Q1R)
      D2=DR**2+DI**2
      XBAR=XBAR-2.*(DR*B(2)+DI*B(5))/D2
      YBAR=ABSF(YBAR-2.*(DR*B(5)-DI*B(2))/D2)
      GO TO 23
  101 RETURN
    1 FORMAT(1H050X,19HLAGUERRE ITERATIONS/31X,9HREAL PART10X,10HIMAG. P
     1ART22X,1HP11X,7HP PRIME6X,11HP DBL PRIME)
      END
C051/5COMPLEX SQUARE ROOT
      SUBROUTINE CXSQRT(A,B,X,Y)
      F=MAX1F(ABSF(A),ABSF(B))
      F=F*SQRTF((A/F)**2+(B/F)**2)
      IF(A)1,1,2
    1 Y=SQRTF((F-A)*.5)
      X=.5*B/Y
      IF(X)4,3,3
    4 X=-X
      Y=-Y
      GO TO 3
    2 X=SQRTF((F+A)*.5)
      Y=.5*B/X
    3 RETURN
      END
      SUBROUTINE SCALE (D,B,M)
      DIMENSION B(6)
      S = 2. ** (XINTF (LOGF(D)/.69314718056) + 1)
      DO 1 I=1,M
    1 B(I) = B(I)/S
      RETURN
      END
```

```
          IDENT     OVFSET
          ENTRY     OVFSET,OVFTST,OVFLU
 OVFSET   SLJ       **
          NOP
+         CALL      SELECT*
          11        OVFLU
          CALL      ERROR*
+         ENA       0
          STA       =SZIG
          SLJ       OVFSET
 OVFLU    SLJ       **
          LDA       =D1.
          STA       ZIG
+         CALL      SELECT*
          11        OVFLU
          CALL      ERROR*
+         SLJ       OVFLU
 OVFTST   SLJ       **
          LDA       *
+         ARS       24
          SAL       *+2
+         INA       1
          SAU       OVFTST
+         LDA       ZIG
          STA       **
          ENA       0
          STA       ZIG
          SLJ       OVFTST
          END
          END
          FINIS
-EXECUTER.
2.0 0.0   10  12 1.0 1.0 1.0
2.0 0.0    9  13 1.0 1.0 1.0
2.0 0.0    9  11 1.0 1.0 1.0
2.0 0.0    8  10 1.0 1.0 1.0
2.0 0.0    7   9 1.0 1.0 1.0
2.0 0.0    6   8 1.0 1.0 1.0
```

```
                    SUM OF EIGENVALUES = -2.26462293E 00

                               TRACE = -1.64628978E 01

                               ALMOST TRIANGULAR FORM

                               TRACE = -1.64628978E 01

                                 LAGUERRE ITERATIONS
               REAL PART          IMAG. PART              P           P PRIME       P DBL PRIME
ITERATE     -3.62580641E 00           0             4.4767E142    1.0649E144    2.4920E145
ITERATE     -3.01794003E 00           0             1.7609E134    3.0870E136    1.5365E138
ITERATE     -3.01119792E 00           0             4.2412E128    2.1767E136    1.1779E138

EIGENVALUE  -3.01119792E 00           0              3 ITERATIONS,TEST 1


                    SUM OF EIGENVALUES = -3.01119792E 00

                               TRACE = -1.62211422E 01

                               ALMOST TRIANGULAR FORM

                               TRACE = -1.62211422E 01

                                 LAGUERRE ITERATIONS
               REAL PART          IMAG. PART              P           P PRIME       P DBL PRIME
ITERATE     -3.34588944E 00           0             6.6601E130    1.6343E132    3.9373E133
ITERATE     -2.83056359E 00           0             1.7088E123    4.2148E125    2.1578E127
ITERATE     -2.82600877E 00           0             2.0802E117    3.3155E125    1.8007E127

EIGENVALUE  -2.82600877E 00           0              3 ITERATIONS,TEST 1
```

```
SUM OF EIGENVALUES = -2.82600877E 00

TRACE = -1.87826649E 01

ALMOST TRIANGULAR FORM

TRACE = -1.87826649E 01

                              LAGUERRE ITERATIONS
               REAL PART          IMAG. PART             P           P PRIME       P DBL PRIME
ITERATE        -3.71464573E 00        0             8.8678E153    2.1799E155    5.2737E156
ITERATE        -3.14332181E 00        0             4.9362E145    1.1206E148    5.7555E149
ITERATE        -3.13831351E 00        0             5.8275E139    8.5935E147    4.7085E149

EIGENVALUE     -3.13831351E 00        0              3 ITERATIONS,TEST 1


SUM OF EIGENVALUES = -3.13831351E 00

TRACE = -4.10898766E 00

ALMOST TRIANGULAR FORM

TRACE = -4.10898766E 00

                              LAGUERRE ITERATIONS
               REAL PART          IMAG. PART             P           P PRIME       P DBL PRIME
ITERATE        -1.06089088E 00        0             1.2649E 27    4.0066E 28    1.1074E 30
ITERATE        -9.62604185E-01        0             3.2971E 23    1.3668E 27    6.6192E 28
ITERATE        -9.62361542E-01        0             5.7849E 15    1.3508E 27    6.5641E 28

EIGENVALUE     -9.62361542E-01        0              3 ITERATIONS,TEST 1


SUM OF EIGENVALUES = -9.62361542E-01

TRACE = -2.28281472E 01

ALMOST TRIANGULAR FORM

TRACE = -2.28281472E 01

                              LAGUERRE ITERATIONS
               REAL PART          IMAG. PART             P           P PRIME       P DBL PRIME
ITERATE        -4.50812887E 00        0             2.8679E193    6.9266E194    1.6524E196
ITERATE        -3.76686923E 00        0             2.5228E183    3.9663E185    1.9965E187
ITERATE        -3.75915319E 00        0             4.8061E177    2.6414E185    1.4623E187

EIGENVALUE     -3.75915319E 00        0              3 ITERATIONS,TEST 1


SUM OF EIGENVALUES = -3.75915319E 00
```

ASPECT RATIO=3.0000E 00/4.0000E 00

K1= 4.2836819345E 01
K2= -1.0960979962E 00
K3= 8.6295336019E-02
K4= -1.5853260079E 00
K5= 4.6966760219E-02
K6= -5.2478409571E-02

6.00000000000E 00	1.19999999998E 01	4.23911630723E 01	4.28368193435E 01
6.00000000000E 00	1.19999999998E 01	4.25079996150E 01	4.28368193458E 01
9.00000000000E 00	1.00000000000E 01	4.24627132912E 01	4.28368193458E 01
1.00000000000E 01	1.00000000000E 01	4.24856640166E 01	4.28368193458E 01
6.00000000000E 00	5.00000000000E 00	4.15644206991E 01	4.28368193435E 01
1.00000000000E 01	1.19999999998E 01	4.25627772359E 01	4.28368193458E 01

EIGENVECTORS

1.93180634477E 00	1.90206795227E 00	3.67451314861E 00	2.61797171802E 00	5.05753312854E 00
3.07761031127E 00	5.94548665138E 00	3.23599098396E 00	6.25145457429E 00	3.07761031244E 00
5.94548659073E 00	2.61797169916E 00	5.05753296474E 00	1.90206777555E 00	3.67451278091E 00
9.99976206524E-01	1.93180568962E 00	2.73198617942E 00	3.34598583577E 00	5.19654616516E 00
6.36444312183E 00	7.15243194951E 00	8.75990430266E 00	8.40818790137E 00	1.02978850124E 01
8.84089188231E 00	1.08278370101E 01	8.40818779776E 00	1.02978849024E 01	7.15243172750E 00
8.75990409381E 00	5.19654581975E 00	6.36444285279E 00	2.73198581033E 00	3.34598561510E 00
3.73196217231E 00	3.86361149489E 00	7.09861382109E 00	7.34902574739E 00	9.77040364104E 00
1.01150661215E 01	1.14857982614E 01	1.18909733335E 01	1.20768829290E 01	1.25029092227E 01
1.14857981645E 01	1.18909732538E 01	9.77040346712E 00	1.01150659868E 01	7.09861362085E 00
7.34902559954E 00	3.73196202889E 00	3.86361139675E 00	3.73196210130E 00	3.34598567069E 00
7.09861373331E 00	6.36444294092E 00	9.77040357329E 00	8.75990417926E 00	1.14857982233E 01
1.02978849522E 01	1.20768829127E 01	1.08278369909E 01	1.14857981625E 01	1.02978849041E 01
9.77040347038E 00	8.75990410289E 00	7.09861362493E 00	6.36444286222E 00	3.73196203180E 00
3.34598562145E 00	2.73198585457E 00	1.93180572128E 00	5.19654589251E 00	3.67451283528E 00
7.15243180259E 00	5.05753302528E 00	8.40818784712E 00	5.94548664091E 00	8.84089187765E 00
6.25145459676E 00	8.40818781336E 00	5.94548661995E 00	7.15243174788E 00	5.05753299047E 00
5.19654583710E 00	3.67451280018E 00	2.73198581976E 00	1.93180569963E 00	9.99976223544E-01
1.90206780558E 00	2.61797173525E 00	3.07761034928E 00	3.23599101778E 00	3.07761033904E 00
2.61797171854E 00	1.90206778850E 00	9.99976213020E-01		

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library U. S. Naval Postgraduate School Monterey, California	2
3.	Naval Ship Systems Command Department of the Navy Washington, D. C. 20360	1
4.	FOIC, Philippine Navy Headquarters Dewey Blvd., Manila, Philippines	1
5.	Prof. R. E. Newton Department of Mechanical Engineering U. S. Naval Postgraduate School Monterey, California	1
6.	LT Maximo S. Dumlao 145 K 1st Kamuning Quezon City, Philippines	1
7.	Prof. G. Cantin Department of Mechanical Engineering U. S. Naval Postgraduate School Monterey, California	1
8.	Mr. John F. Yardley McDonnell Aircraft Corp., St Louis, Missouri 63166	1

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
U. S. NAVAL POSTGRADUATE SCHOOL, Monterey, Calif.

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
N.A.

3. REPORT TITLE
Finite difference solutions for plate buckling problems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
N.A.

5. AUTHOR(S) *(Last name, first name, initial)*
DUMLAO, Maximo S., JR, LT, Philippine Navy

6. REPORT DATE
May 66

7a. TOTAL NO. OF PAGES
85

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.
N.A.

b. PROJECT NO.
N.A.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
N.A.

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
N.A.

10. AVAILABILITY/LIMITATION NOTICES
[redacted]
9/15/69 GRL
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
U. S. Navy

13. ABSTRACT

DD FORM 1 JAN 64 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Buckling of rectangular plates Clamped and simply supported. Small deflection theory. Digital computer program.						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8c, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.